The Formal Language PL

vocabulary

atomic sentence symbols: ‘p’, ‘q’, ‘r’, ‘s’, ‘p_1’, ‘q_1’, ‘r_1’, ‘s_1’, ‘p_2’, ...

connectives: ‘$\neg$’ (negation connective)

‘$\&$’ (conjunction connective)

‘$\vee$’ (disjunction connective)

‘$\leftrightarrow$’ (biconditional connective)

‘$\rightarrow$’ (conditional connective)

grouping symbols: ‘(’ , ‘)’

<u>definition</u> : An uninterrupted, left-to-right, finite sequence of vocabulary symbols is a <u>PL-string</u>.

rules of formation

(i) An atomic sentence symbol standing alone is a sentence of PL — an atomic sentence.

(ii) If A and B are sentences of PL, then all of the strings $\ulcorner \neg A \urcorner$, $\ulcorner (A \& B) \urcorner$, $\ulcorner (A \vee B) \urcorner$, $\ulcorner (A \leftrightarrow B) \urcorner$, and $\ulcorner (A \rightarrow B) \urcorner$ are sentences of PL.

(iii) No other PL-strings are sentences of PL

examples

of sentences: ‘p’, ‘$(p \rightarrow q)$’, ‘$((p \& q) \leftrightarrow (\neg p \vee \neg r))$’

of non-sentences: ‘pqr’, ‘$\& pq$’, ‘$(A \rightarrow B)$’

rules of interpretation

(i) Each atomic sentence is either true or false.

(ii) A sentence $\ulcorner \neg A \urcorner$ is true if and only if A is false; otherwise, it is false.

(iii) A sentence $\ulcorner (A \;\&\; B) \urcorner$ is true if and only if both A and B are true; otherwise, it is false.

(iv) A sentence $\ulcorner (A \vee B) \urcorner$ is true if and only if either A or B or both are true; otherwise, it is false.

(v) A sentence $\ulcorner (A \leftrightarrow B) \urcorner$ is true if and only if either both A and B are true or both A and B are false; otherwise, it is false.

(vi) A sentence $\ulcorner (A \rightarrow B) \urcorner$ is true if and only if either A is false or B is true (or both); otherwise, it is false.

tables summarizing the rules

A	B	$\ulcorner \neg A \urcorner$	$\ulcorner \neg B \urcorner$	$\ulcorner (A \;\&\; B) \urcorner$	$\ulcorner (A \vee B) \urcorner$	$\ulcorner (A \leftrightarrow B) \urcorner$	$\ulcorner (A \rightarrow B) \urcorner$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	F
F	T	T	F	F	T	F	T
F	F	T	T	F	F	T	T

Validity of PL Sentence Schemata

__definition__: A PL sentence schema is __valid__ if and only if every instance of it is true regardless of the truth-values of the atomic sentences of PL.

__theorem__: A PL sentence schema is __valid__ if and only if it is tautologous (i.e. all the entries in the governing column of its truth-table are __T__'s).

__decision procedure__

1. construct the truth-table
2. if all __T__'s, then schema is valid; otherwise, not.

__example__

$\ulcorner (A \to (B \to A)) \urcorner$

1.

A	B	$(A \to (B \to A))$	
T	T	T	t
T	F	T	t
F	T	T	f
F	F	T	t

2. yes

Logical Truth

<u>definition</u>: A PL sentence is a <u>logical truth</u> of PL if and only if it is true regardless of the truth-values of the atomic sentences of PL.

<u>theorem</u>: A PL sentence is a logical truth of PL if and only if it is a tautology (i.e. an instance of a tautologous schema).

<u>decision procedure</u>

1. form the principal schema of the sentence — i.e. the schema in which the atomic sentences are replaced by schematic letters, each occurrence of any one atomic sentence being replaced by the same schematic letter {and different atomic sentences being replaced by different schematic letters
2. construct the truth-table of this schema
3. if all <u>T</u>'s, then sentence is logically true; otherwise, not.

<u>example</u>

'$((p \vee \neg p) \vee q)$'

1., 2.

A	B	$((A \vee \neg A) \vee B)$	
T	T	t	T
T	F	t	T
F	T	t	T
F	F	t	T

3. yes

Truth-functional Equivalence

<u>definition</u>: Two PL sentences are <u>truth-functionally equivalent</u> if and only if they have the same truth-value regardless of the truth-values of the atomic sentences of PL.

<u>theorem</u>: Two PL sentences, A and B, are truth-functionally equivalent if and only if the PL sentence $\ulcorner(A \leftrightarrow B)\urcorner$ is a logical truth of PL.

<u>decision procedure</u>

1. form the biconditional sentence
2. form the principal schema of this sentence
3. construct the truth-table of this schema
4. if all <u>T</u>'s, then truth-functionally equivalent; otherwise, not

<u>example</u>

'$(p \rightarrow q)$' and '$(\neg q \rightarrow \neg p)$'

1. '$((p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p))$'
2. $\ulcorner((A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A))\urcorner$
3.

A	B	$((A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A))$
T	T	t T f t f
T	F	f T t f f
F	T	t T f t t
F	F	t T t t t

4. yes, '$(p \rightarrow q)$' and '$(-q \rightarrow -p)$' are truth-functionally equivalent

Truth-functional Consequence

<u>definition</u>: A PL sentence B is a <u>truth-functional consequence</u> of a PL sentence A if and only if, regardless of the truth-values of the atomic sentences of PL, it is not the case that B is false and A is true.

<u>theorem</u>: A PL sentence B is a truth-functional consequence of a PL sentence A if and only if the PL sentence $\ulcorner(A \rightarrow B)\urcorner$ is a logical truth of PL.

<u>decision procedure</u>

1. form the conditional sentence
2. form the principal schema of this sentence
3. construct the truth-table of this schema
4. if all <u>T</u>'s, then a truth-functional consequence; otherwise, not

<u>example</u>

'$(q \rightarrow p)$' and 'p'

1. '$(p \rightarrow (q \rightarrow p))$'
2. $\ulcorner(A \rightarrow (B \rightarrow A))\urcorner$
3.

A	B	(A → (B → A))	
T	T	T	t
T	F	T	t
F	T	T	f
F	F	T	t

4. yes, '$(q \rightarrow p)$' is a truth-functional consequence of 'p'.

Validity of PL Arguments

<u>definition</u>: A PL argument is <u>valid</u> if and only if, regardless of the truth-values of the atomic sentences of PL, it is not the case that the conclusion of the argument is false and all of the premisses of the argument are true.

<u>theorem</u>: A PL argument is valid if and only if the PL conditional sentence, in which the conclusion of the argument is the consequent and the conjunction of the premisses is the antecedent, is a logical truth of PL.

<u>decision procedure</u>

1. form conjunction of the premisses
2. form conditional, with conjunction of premisses as antecedent and conclusion as consequent
3. use truth-table technique to determine whether this sentence is a logical truth
4. if a logical truth, then argument valid; otherwise, invalid

<u>example</u>

$(p \vee q)$

$\neg p$

$\therefore q$

1. $((p \vee q) \,\&\, \neg p)$
2. $(((p \vee q) \,\&\, \neg p) \rightarrow q)$
3.

A	B	$(((A \vee B) \,\&\, \neg A) \rightarrow B)$
T	T	t f f T
T	F	t f f T
F	T	t t t T
F	F	f f t T

Validity of PL Argument Schemata

<u>definition</u>: A PL argument schema is <u>valid</u> if and only if all of its instances are valid.

<u>theorem</u>: A PL argument schema is valid if and only if the PL conditional sentence schema, in which the conclusion of the argument is the consequent and the conjunction of the premisses is the antecedent, is a valid sentence schema of PL.

<u>decision procedure</u>

1. form the sentence schema consisting of the conjunction of the premisses.
2. form the conditional sentence schema, with the conjunction of the premisses as antecedent and the conclusion as consequent.
3. construct the truth-table of this schema
4. if all <u>T</u>'s, then argument schema is valid; otherwise, not

<u>example</u>

$(A \to B)$

A

$\therefore B$

1. $\ulcorner((A \to B) \& A)\urcorner$
2. $\ulcorner(((A \to B) \& A) \to B)\urcorner$
3.

A	B	(((A → B) & A) → B)		
T	T	t	t	T
T	F	f	f	T
F	T	t	f	T
F	F	t	f	T

4. yes, valid

Two Theorems About PL

1. Suppose A is a logical truth of PL and B is a truth-functional consequence of A. Then B is a logical truth of PL.

2. Suppose B is truth-functionally equivalent to C, and suppose A and A′ are alike save for A′ having an occurrence of C where A has an occurrence of B. Then A and A′ are truth-functionally equivalent, so that A′ is a logical truth of PL if and only if A is.

Remarks

1. Both of these principles enable one to infer or prove that various sentences of PL are logically true from the previously established fact that certain other sentences are logically true.

2. If you already know that A is a logical truth and that B is a truth-functional consequence of it — i.e. that $\ulcorner(A \rightarrow B)\urcorner$ is a logical truth — then you can infer that B is a logical truth without having to construct a truth-table for it.

3. Say you want to decide whether A′ is a logical truth. Suppose A′ contains an occurrence of C, and suppose that you already know that C is truth-functionally equivalent to B — i.e. that $\ulcorner(B \leftrightarrow C)\urcorner$ is a logical truth. Then it may help to look at A, where A is obtained from A′ by <u>substituting</u> B for the occurrence of C. In particular, it will help when you have previously established whether A is a logical truth, for you will then be able to infer directly whether A′ is a logical truth.

Some Laws of PL

The following schemata are tautologous

$(A \vee \neg A)$ – law of excluded middle

$\neg(A \,\&\, \neg A)$ – law of contradiction

$$\left.\begin{array}{l}(A \rightarrow A)\\ (A \leftrightarrow A)\end{array}\right\}\ \text{laws of reflexivity}$$

$$\left.\begin{array}{l}((A \,\&\, B) \leftrightarrow (B \,\&\, A))\\ ((A \vee B) \leftrightarrow (B \vee A))\\ ((A \leftrightarrow B) \leftrightarrow (B \leftrightarrow A))\end{array}\right\}\ \text{laws of symmetry}$$

$$\left.\begin{array}{l}(((A \,\&\, B) \,\&\, C) \leftrightarrow (A \,\&\, (B \,\&\, C)))\\ (((A \vee B) \vee C) \leftrightarrow (A \vee (B \vee C)))\end{array}\right\}\ \text{associative laws}$$

hence can drop parentheses within a sequence of conjunctions and within a sequence of disjunctions — e.g. writing ⌜$(A \,\&\, B \,\&\, C)$⌝ as an abbreviation for ⌜$((A \,\&\, B) \,\&\, C)$⌝

$$\left.\begin{array}{l}(((A \vee B) \,\&\, C) \leftrightarrow ((A \,\&\, C) \vee (B \,\&\, C)))\\ (((A \,\&\, B) \vee C) \leftrightarrow ((A \vee C) \,\&\, (B \vee C)))\end{array}\right\}\ \text{distributive laws}$$

$((A \rightarrow B) \leftrightarrow (\neg A \vee B))$ – law of the material conditional

$((A \leftrightarrow B) \leftrightarrow ((\neg A \vee B) \,\&\, (A \vee \neg B)))$ – law of the material biconditional

$(A \leftrightarrow \neg\neg A)$ – law of double negation

$(A \leftrightarrow (A \,\&\, (B \vee \neg B)))$ – law of tautologies

$(A \leftrightarrow (A \vee (B \,\&\, \neg B)))$ – law of contradictories

$$\left.\begin{array}{l}(\neg(A \,\&\, B) \leftrightarrow (\neg A \vee \neg B))\\ (\neg(A \vee B) \leftrightarrow (\neg A \,\&\, \neg B))\end{array}\right\}\ \text{de Morgan's laws}$$

$((A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)))$ – law of transitivity of '$\rightarrow$'

$(A \rightarrow (B \rightarrow A))$ – law of the weak conditional

Disjunctive Normal Form

<u>definition</u>: A PL sentence is in <u>disjunctive normal form</u> just in case it consists of

(i) an atomic sentence or the negation of an atomic sentence

(ii) <u>or</u> a conjunction (with two or more conjuncts), each conjunct of which is an atomic sentence or the negation of an atomic sentence.

(iii) <u>or</u> a disjunction (with two or more disjuncts), each disjunct of which satisfies either (i) or (ii) preceding.

<u>examples</u>

'p', '$(p \& q)$', '$(p \vee (\neg p \& \neg q) \vee (\neg p \& q \& r \& \neg s))$'

<u>theorem</u>: Corresponding to each sentence of PL is a truth-functionally equivalent sentence of PL in disjunctive normal form.

<u>example of a derivation</u>

$(((p \rightarrow q) \rightarrow p) \rightarrow p)$	
$(\neg(\neg(\neg p \vee q) \vee p) \vee p)$	via law of material conditional
$((\neg\neg(\neg p \vee q) \& \neg p) \vee p)$	via de Morgan law
$(((\neg p \vee q) \& \neg p) \vee p)$	via law of double negation
$(((\neg p \& \neg p) \vee (q \& \neg p)) \vee p)$	via distributive law
$((\neg p \& \neg p) \vee (q \& \neg p) \vee p)$	via associative law
$(\neg p \vee (q \& \neg p) \vee p)$	via simplification : $\ulcorner((A \& A) \leftrightarrow A)\urcorner$

<u>theorem</u>: Corresponding to each sentence of PL is a truth-functionally equivalent sentence of PL′ — i.e. of the fragment of PL in which '$\neg$', '$\&$', and '$\vee$' are the only connectives.

Conjunctive Normal Form

<u>definition</u>: A PL sentence is in <u>conjunctive normal form</u> just in case it consists of

(i) an atomic sentence or the negation of an atomic sentence

(ii) <u>or</u> a disjunction (with two or more disjuncts), each disjunct of which is an atomic sentence or the negation of an atomic sentence.

(iii) <u>or</u> a conjunction (with two or more conjuncts), each conjunct of which satisfies either (i) or (ii) preceding

<u>examples</u>

'p' , '$(p \vee q)$' , '$(p \,\&\, (q \vee r) \,\&\, (\neg p \vee \neg q \vee \neg r \vee s))$'

<u>theorem</u>: Corresponding to each sentence of PL is a truth-functionally equivolent sentence of PL in conjunctive normal form.

<u>example of a derivation</u>

$(((p \rightarrow q) \rightarrow p) \rightarrow p)$	
$(\neg(\neg(\neg p \vee q) \vee p) \vee p)$	via law of material conditional
$((\neg\neg(\neg p \vee q) \,\&\, \neg p) \vee p)$	via de Morgan law
$(((\neg p \vee q) \,\&\, \neg p) \vee p)$	via law of double negation
$(((\neg p \vee q) \vee p) \,\&\, (\neg p \vee p))$	via distributive law
$((\neg p \vee q \vee p) \,\&\, (\neg p \vee p))$	via ossociative law

<u>theorem</u>: Suppose A is a sentence of PL and B is a sentence of PL in conjunctive normal form that is truth-functionally equivalent to A. Then A is a logical truth of PL if and only if each conjunct of B includes both an atomic sentence and its negation.

PLTM

Rules

$$\begin{array}{c} (A \rightarrow B) \\ \diagup \quad \diagdown \\ \neg A \qquad B \end{array} \qquad \begin{array}{c} (A \,\&\, B) \\ | \\ A \\ B \end{array} \qquad \begin{array}{c} (A \vee B) \\ \diagup \quad \diagdown \\ A \qquad B \end{array} \qquad \begin{array}{c} (A \leftrightarrow B) \\ \diagup \quad \diagdown \\ \begin{array}{c} A \\ B \end{array} \qquad \begin{array}{c} \neg A \\ \neg B \end{array} \end{array}$$

$$\begin{array}{c} \neg(A \rightarrow B) \\ | \\ A \\ \neg B \end{array} \qquad \begin{array}{c} \neg(A \,\&\, B) \\ \diagup \quad \diagdown \\ \neg A \qquad \neg B \end{array} \qquad \begin{array}{c} \neg(A \vee B) \\ | \\ \neg A \\ \neg B \end{array} \qquad \begin{array}{c} \neg(A \leftrightarrow B) \\ \diagup \quad \diagdown \\ \begin{array}{c} \neg A \\ B \end{array} \qquad \begin{array}{c} A \\ \neg B \end{array} \end{array}$$

Definition: A sentence is a theorem of PLTM if and only if every path in every tree obtained by applying the tree method of PLTM to it is closed.

Definition: A sentence B is a formal consequence in PLTM of a set of sentences $\{A_1, \ldots, A_n\}$ if and only if every path in every tree obtained by applying the tree method of PLTM to the argument with $A_1, \ldots, A_n$ as premisses and B as conclusion. is closed.

Theorem: Any tree obtained by applying the tree method of PLTM to a sentence or argument has an open path if and only if every tree obtained by applying the method to this sentence or argument has an open path.

Theorem: A sentence of PL is a logical truth if and only if it is a theorem of PLTM.

Theorem: An argument of PL is valid if and only if its conclusion is a formal consequence in PLTM of its premisses.

Theorem: The tree method of PLTM provides a decision procedure for logical truthhood and argument validity in PL.

Proof: The tree method always terminates finitely in the case of PLTM.