DE GRUYTER OPEN

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 42 (55) 2015
DOI: 10.1515/slgr-2015-0032

Lei Ma
Huaqiao University & Southeast University, China

TRUTH-GRAPH METHOD: A HANDY METHOD DIFFERENT FROM THAT OF LEŚNIEWSKI'S

Abstract. The paper presents a method of truth-graph by truth-tables. On the one hand, the truth-graph constituted by truth value coordinate and circumference displays a more visual representation of the different combinations of truth-values for the simple or complex propositions. Truth-graphs make sure that you don't miss any of these combinations. On the other hand, they provide a more convenient tool to discern the validity of a complex proposition made up by simple compositions. The algorithm involving in setting up all the truth conditions is proposed to distinguish easily among tautologous, contradictory and consistent expressions. Furthermore, the paper discusses a certain connection between the truth graphs and the symbols for propositional connectives proposed by Stanisław Leśniewski.

Keywords: truth-graphs, truth-tables, truth value coordinate, algorithm, Stanisław Leśniewski.

1. Introduction

In the history of logic we can express extensive relations among concepts by Euler Diagrams and Venn Diagrams, which are convenient for some fundamental calculations by intuition. However, we have not met analogous diagrams, since the birth of propositional logic, to help us comprehend and grasp the fundamental logical relations among propositions by intuition and make some basic calculations among propositions. C.S. Peirce, an American logician and philosopher, advanced a method for a truth-table, which is a powerful tool for the study of propositional logic, through permuting every possible combination of acquiring truth values by every propositional variable of a truth expression into a table (Peirce, 1931–1935). The method, with which we study and distinguish whether or not a truth expression is true and evaluate its truth value, is called the method of truth-table. It is sometimes troublesome, however, to write and permute. Moreover, this method appears quite clumsy for finding out the simplest equivalent expression of a complex truth expression. If we turn the truth-tables into truth-graphs and develop the method of a truth-graph, we shall make things much easier by intuition. In addition, it is interesting to compare the truth-graph with the notation introduced by Stanisław Leśniewski. There are certain connections and differences between the two signs, leading to different geometric representation and application for evaluating propositional connectives.

2. Six Basic Truth-graphs

There are six basic truth-graphs, matched with six basic truth-tables, in propositional logic. Considering propositional logic, which usually uses the truth-tables to show the truth value relations between two propositions, I present a handy graph method, which can intuitively reflect the truth value relations between propositions and determine the truth value of a compound proposition, to depict the truth-tables. For easier comparison, I draw the basic truth-tables and their relevant truth-graphs as follows.

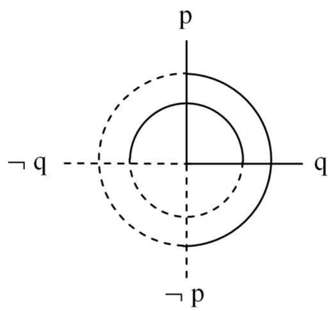
2.1. Negative Expression

Its truth-table is:

	p(q)	¬p(¬q)
1)	T	F
2)	F	T

Table 1: Truth-table of negative expression

Its truth-graph is:



Graph 1: Truth-graph of negative expression

Table 1 shows that if ‘p(q)’ is true, then ‘¬p(¬q)’ is false, and if ‘p(q)’ is false, then ‘¬p(¬q)’ is true (Greenstein, 1978; Newton-Smith, 1985; Quine, 1982). In graph 1, the four line segments represent propositional variables. Shape ‘+’, which is constituted by two solid lines and two dotted lines, may be named a truth-value coordinate with propositional variables ‘p’ (by the solid line, in the upper part) and ‘¬p’ (by the dotted line, in the lower part), ‘q’ (by the solid line, on the right) and ‘¬q’ (by the dotted line, on the left). The inner semi-circumference (by the solid line, in the upper part) and the other inner semi-circumference (by the dotted line, in the lower part) signify that if ‘p’ is true, then ‘¬p’ is false, and, ‘p’ is true, regardless of the truth value of ‘q’. Similarly, the outer semi–circumference (by the solid line, on the right) and the other outer semi-circumference (by the dotted line, on the left) signify that if ‘q’ is true, then ‘¬q’ is false, and, ‘q’ is true, regardless of the truth value of ‘p’. In the graph, the inner circumference stands for truth-graph ‘p’, the outer stands for truth-graph ‘q’. If we convert the solid line of the inner semi-circumference into a dotted line, and the dotted line into a solid line, we will have the truth-graph ‘¬p’, which means ‘¬p’ is true whether ‘q’ is true or not, and if ‘¬p’ is true, then ‘p’ is false. So does truth-graph ‘¬q’. From the graph, we can see that the truth-value of ‘p’ or ‘¬p’ has nothing to do with that of ‘q’ or ‘¬q’, and that it is contradictory between ‘p’ and ‘¬p’, and ‘q’ and ‘¬q’.

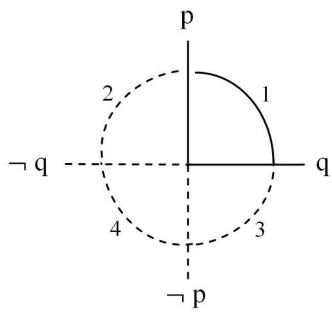
2.2. Conjunctive Expression

Its truth-table is:

	p	q	p∧q
1)	T	T	T
2)	T	F	F
3)	F	T	F
4)	F	F	F

Table 2: Truth-table of conjunctive expression

Its truth-graph is:



Graph 2: Truth-graph of conjunctive expression

Table 2 shows that ‘p∧q’ is true if and only if both ‘p’ and ‘q’ are true (Greenstein, 1978; Newton-Smith, 1985; Quine, 1982). In graph 2, the circumference stands for the truth expression ‘p∧q’, and this circumference is separated by the lines into arcs, 1, 2, 3, and 4, each representing a truth value respectively in the conjunctive truth-table. Take two examples, arc 1 represents that ‘p∧q’ is true if both ‘p’ and ‘q’ are true, arc 2 represents that ‘p∧q’ is false if ‘p’ is true and ‘q’ is false.

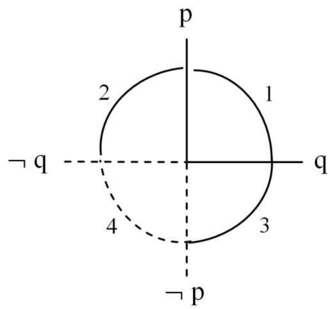
2.3. Disjunctive Expression

Its truth-table is:

	p	q	p∨q
1)	T	T	T
2)	T	F	T
3)	F	T	T
4)	F	F	F

Table 3: Truth-table of disjunctive expression

Its truth-graph is:



Graph 3: Truth-graph of disjunctive expression

Table 3 shows that ‘p∨q’ is true if and only if ‘p’, ‘q’ are not all false (Greenstein, 1978; Newton-Smith, 1985; Quine, 1982). In graph 3, the circumference stands for the truth expression ‘p∨q’, and this circumference is separated by the lines into four arcs, 1, 2, 3, and 4, each representing a truth value respectively in the disjunctive truth-table. For instance, arc 2 represents that ‘p∨q’ is true if ‘p’ is true and ‘q’ is false; arc 4 represents that ‘p∨q’ is false if both ‘p’ and ‘q’ are false.

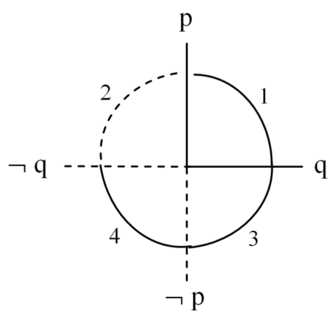
2.4. Implicative Expression

Its truth-table is:

	p	q	p→q
1)	T	T	T
2)	T	F	F
3)	F	T	T
4)	F	F	T

Table 4: Truth-table of implicative expression

Its truth-graph is:



Graph 4: Truth-graph of implicative expression

Table 4 shows that ‘p→q’ is false if and only if ‘p’ is true and ‘q’ is false (Greenstein, 1978; Newton-Smith, 1985; Quine, 1982). In graph 4, the circumference stands for the truth expression ‘p→q’, and this circumference is separated by the lines into four arcs, 1, 2, 3, and 4, each representing a truth value respectively in the implicative truth-table. For example, arc 2 represents that 'p→q' is false if 'p' is true and 'q' is false; arc 3 represents that 'p→q' is true if 'p' is false and 'q' is true.

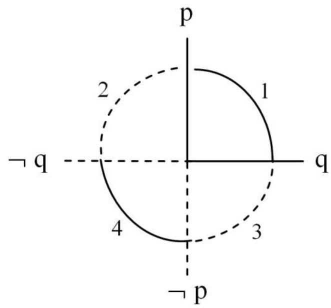
2.5. Equivalent Expression

Its truth-table is:

	p	q	p↔q
1)	T	T	T
2)	T	F	F
3)	F	T	F
4)	F	F	T

Table 5: Truth-table of equivalent expression

Its truth-graph is:



Graph 5: Truth-graph of equivalent expression

Table 5 shows that 'p↔q' is true if and only if both 'p' and 'q' are true or false (Greenstein, 1978; Newton-Smith, 1985; Quine, 1982). In graph 5, the circumference stands for the truth expression 'p↔q', and this circumference is separated by the lines into four arcs, 1, 2, 3, and 4, each representing a truth value respectively in the equivalent table. Take two examples, arc 1 represents that 'p↔q' is true if both 'p' and 'q' are true; arc 3 represents that 'p↔q' is false if 'p' is false and 'q' is false.

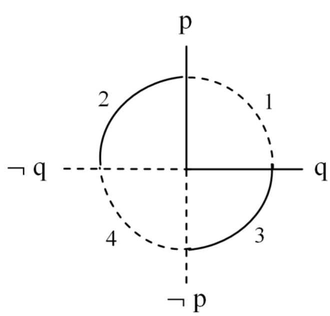
2.6. Incompatible Disjunctive Expression

Its truth-table is:

	p	q	p∀q
1)	T	T	F
2)	T	F	T
3)	F	T	T
4)	F	F	F

Table 6: Truth-table of incompatible disjunctive expression

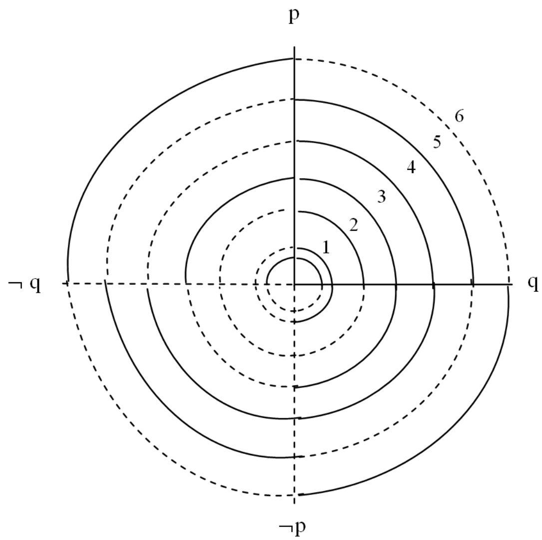
Its truth-graph is:



Graph 6: Truth-graph of incompatible disjunctive expression

Table 6 shows that ‘p∇q’ is false if and only if both ‘p’ and ‘q’ are true or false (Greenstein, 1978; Newton-Smith, 1985; Quine, 1982). In graph 6, the circumference stands for the truth-graph of ‘p∇q’, and this circumference is separated by the lines into four arcs, 1, 2, 3, and 4, each representing a truth value respectively in the equivalent table. For example, arc 2 represents that ‘p∇q’ is true if ‘p’ is true and ‘q’ is false; arc 4 represents that ‘p∇q’ is false if both ‘p’ and ‘q’ are false.

For easier comparison, we can draw all the six basic true graphs in an identical truth coordinate:



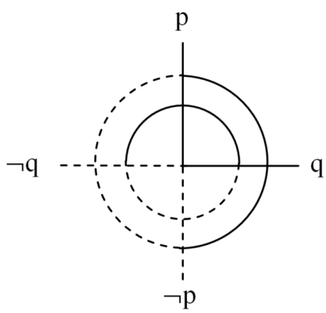
Graph 7: Basic true-graphs in an identical truth coordinate

In graph 7, circumferences 1 are negative truth-graphs, circumference 2 is a conjunctive one, circumference 3 is a compatible disjunctive one, circumference 4 is implicative one, circumference 5 is an equivalent one, and circumference 6 is an incompatible disjunctive one.

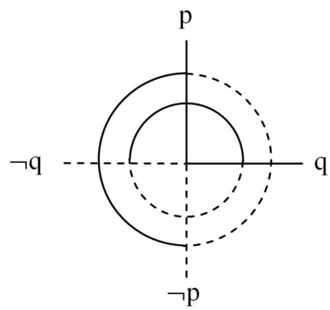
3. Relations among the Six Types of Basic Truth-graphs

For the six basic truth-graphs, if we consider the various truth combinations of propositional variables in every truth-graph, we can draw out the other basic truth-graphs imitatively. In this case we have six types of basic truth-graphs, in which the equivalent graph and the incompatible disjunctive graph has two forms each, while the others have four each.

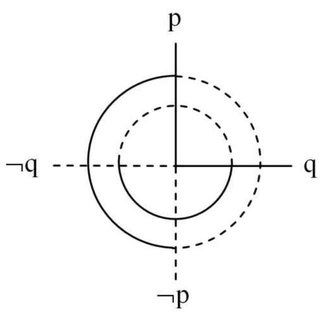
3.1. Negative Graphs



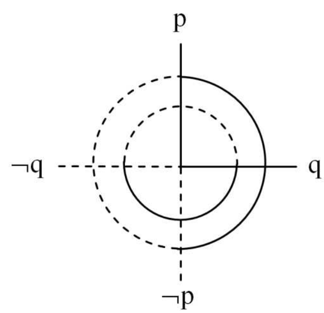
Graph 8(a)



Graph 8(b)



Graph 8(c)



Graph 8(d)

For convenience, I use 'I' to stand for negative graphs, 'II' for conjunctive graphs, 'III' for compatible disjunctive graphs, 'IV' for implicative graphs, 'V' for equivalent graphs, and 'VI' for incompatible disjunctive graphs. In each type, I use, say, 'Ia' to stand for negative graph a, and 'IIb' for conjunctive graph b. The others are followed in such order.

In graph Ia, the inner circumference stands for truth-graph 'p', the outer stands for truth-graph 'q'; in graph Ib, the inner circumference stands for truth-graph 'p', the outer stands for truth-graph '¬q'; in graph Ic, the inner circumference stands for truth-graph '¬p', the outer stands for truth-graph '¬q'; and in graph Id, the inner circumference stands for truth-graph '¬p', the outer stands for truth-graph 'q'.

We can see that there are some relations among the four cases of negative graphs:

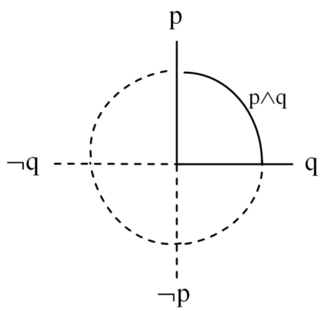
Ia=¬Ic. It means that 'p↔¬¬p', and 'q↔¬¬q', and that we can get 'Ic', by changing solid lines to dotted lines, and dotted lines to solid lines, from 'Ia'.

¬Ia=Ic. It means that ‘¬¬p↔p’, and ‘¬¬q↔q’, and that we can get ‘Ia’, by changing solid lines to dotted lines, and dotted lines to solid lines, from ‘Ic’.

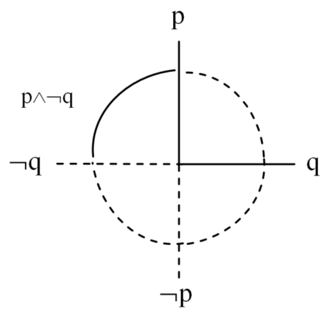
Ib=¬Id. It means that ‘p↔¬¬p’, and ‘¬¬q↔q’, and that we can get ‘Id’, by changing solid lines to dotted lines, and dotted lines to solid lines, from ‘Ib’.

¬Ib=Id. It means that ‘¬¬p↔p’, and ‘q↔¬¬q’, and that we can get ‘Ib’, by changing solid lines to dotted lines, and dotted lines to solid lines, from ‘Id’.

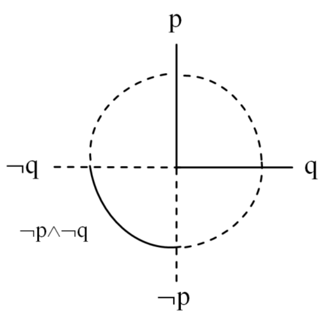
3.2. Conjunctive Graphs



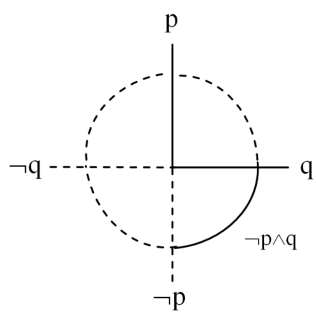
Graph 9(a)



Graph 9(b)



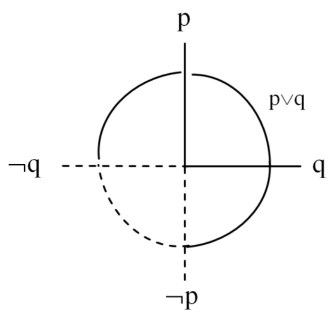
Graph 9(c)



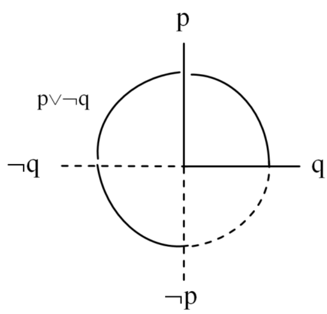
Graph 9(d)

In graph IIa, the circumference stands for truth-graph ‘p∧q’ or ‘q∧p’; in graph IIb, the circumference stands for truth-graph ‘p∧¬q’ or ‘¬q∧p’; in graph IIc, the circumference stands for truth-graph ‘¬p∧¬q’ or ‘¬q∧¬p’; in graph IId, the circumference stands for truth-graph ‘¬p∧q’ or ‘q∧¬p’.

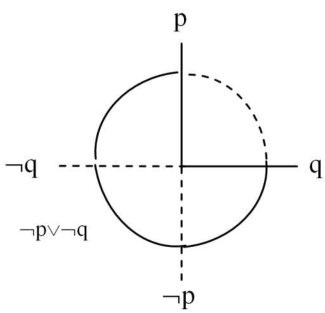
3.3. Disjunctive Graphs



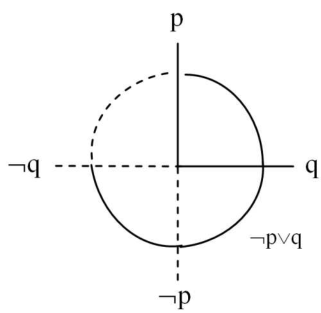
Graph 10(a)



Graph 10(b)



Graph 10(c)



Graph 10(d)

In graph IIIa, the circumference stands for truth-graph '$p \vee q$' or '$q \vee p$'; in graph IIIb, the circumference stands for truth-graph '$p \vee \neg q$' or '$\neg q \vee p$'; in graph IIIc, the circumference stands for truth-graph '$\neg p \vee \neg q$' or '$\neg q \vee \neg p$'; in graph IIId, the circumference stands for truth-graph '$\neg p \vee q$' or '$q \vee \neg p$'.

Contrasting III with II, it is not hard to see the relations among them:

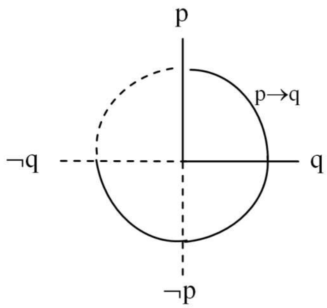
IIa=¬IIIc or ¬IIa=IIIc. It means that '$p \wedge q \leftrightarrow \neg(\neg p \vee \neg q)$' or '$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$', and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'IIIc' from 'IIa', or get 'IIa' from 'IIIc'.

IIb=¬IIId or ¬IIb=IIId. It means that '$p \wedge \neg q \leftrightarrow \neg(\neg p \vee q)$' or '$\neg(p \wedge \neg q) \leftrightarrow (\neg p \vee q)$', and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'IIId' from 'IIb', or get 'IIb' from 'IIId'.

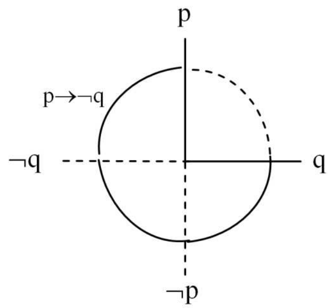
IIc=¬IIIa or ¬IIc=IIIa. It means that '$\neg p \wedge \neg q \leftrightarrow \neg(p \vee q)$' or '$\neg(\neg p \wedge \neg q) \leftrightarrow (p \vee q)$', and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'IIIa' from 'IIc', or get 'IIc' from 'IIIa'.

IId=¬IIIb or ¬IId=IIIb. It means that '$\neg p \wedge q \leftrightarrow \neg(p \vee \neg q)$' or '$\neg(\neg p \wedge q) \leftrightarrow (p \vee \neg q)$', and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'IIIb' from 'IId', or get 'IId' from 'IIIb'.

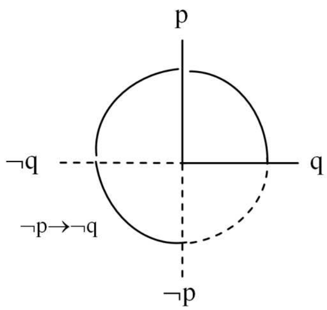
3.4. Implicative Graphs



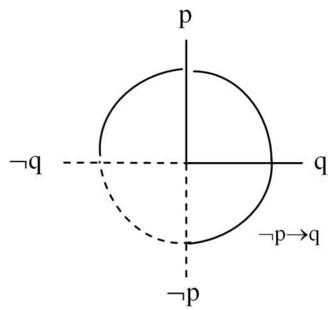
Graph 11(a)



Graph 11(b)



Graph 11(c)



Graph 11(d)

In graph IVa, the circumference stands for truth-graph ‘$p \to q$’ or ‘$\neg q \to \neg p$’; in graph IVb, the circumference stands for truth-graph ‘$p \to \neg q$’ or ‘$q \to \neg p$’; in graph IVc, the circumference stands for truth-graph ‘$\neg p \to \neg q$’ or ‘$q \to p$’; in graph IVd, the circumference stands for truth-graph ‘$\neg p \to q$’ or ‘$\neg q \to p$’.

Contrasting IV with II and III, it is easy to see the relations among them.

Relations among II & IV:

IIa=¬IVb or ¬IIa=IVb. This means that ‘$(p \wedge q) \leftrightarrow \neg(p \to \neg q)$’ or ‘$\neg(p \wedge q) \leftrightarrow (p \to \neg q)$’, and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get ‘IVb’ from ‘IIa’, or get ‘IIa’ from ‘IVb’.

IIb=¬IVa or ¬IIb=IVa. This means that ‘$(p \wedge \neg q) \leftrightarrow \neg(p \to q)$’ or ‘$\neg(p \wedge \neg q) \leftrightarrow (p \to q)$’, and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get ‘IVa’ from ‘IIb’, or get ‘IIb’ from ‘IVa’.

IIc=¬IVd or ¬IIc=IVd. This means that ‘$(\neg p \wedge \neg q) \leftrightarrow \neg(\neg p \to q)$’ or ‘$\neg(\neg p \wedge \neg q) \leftrightarrow (\neg p \to q)$’, and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get ‘IVd’ from ‘IIc’, or get ‘IIc’ from ‘IVd’.

IId=¬IVc or ¬IId=IVc. This means that '(¬p∧q)↔¬(¬p→¬q)' or '¬(¬p∧q)↔(¬p→¬q)', and that we can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'IVc' from 'IId', or get 'IId' from 'IVc'.

Relations among III & IV:

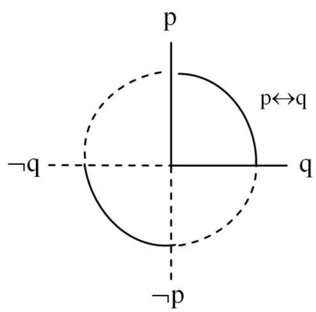
IIIa=IVd. This means that '(p∨q)↔(¬p→q)', and that 'IIIa' and 'IVd' share the same truth-graph.

IIIb=IVc. This means that '(p∨¬q)↔(¬p→¬q)', and that 'IIIb' and 'IVc' share the same truth-graph.

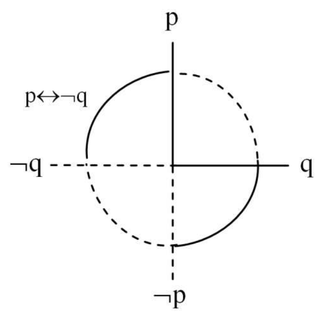
IIIc=IVb. This means that '(¬p∨¬q)↔(p→¬q)', and that 'IIIc' and 'IVb' share the same truth-graph.

IIId=IVa. This means that '(¬p∨q)↔(p→q)', and that 'IIId' and 'IVa' share the same truth-graph.

3.5. Equivalent Graphs



Graph 12(a)



Graph 12(b)

In graph Va, the circumference stands for truth-graph 'p↔q', 'q↔p', '¬p↔¬q', or '¬q↔¬p'; in graph Vb, the circumference stands for truth-graph 'p↔¬q', '¬p↔q', '¬q↔p', or 'q↔¬p'.

We can see the relation between Va & Vb would now appear as follows:

Va=¬Vb or ¬Va=Vb.

This means:

(1) The equivalent expressions 'p↔q', 'q↔p', '¬p↔¬q', and '¬q↔¬p' are all equivalent, and can be represented by the same truth-graph;

(2) The equivalent expressions 'p↔¬q', '¬p↔q', '¬q↔p', and 'q↔¬p' are all equivalent, and can be represented by the same truth-graph;

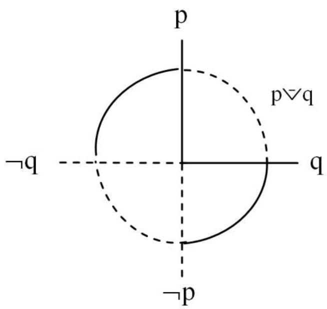
(3) We have these equivalent expressions:
(p↔q)↔¬(p↔¬q), (p↔q)↔¬(¬p↔q), (p↔q)↔¬(¬q↔p),
(p↔q)↔¬(q↔¬p), (q↔p)↔¬(p↔¬q), (q↔p)↔¬(¬p↔q),
(q↔p)↔¬(¬q↔p), (q↔p)↔¬(q↔¬p),
(¬p↔¬q)↔¬(p↔¬q), (¬p↔¬q)↔¬(¬p↔q), (¬p↔¬q)↔¬(¬q↔p),

(¬p↔¬q)↔¬(q↔¬p), (¬q↔¬p)↔¬(p↔¬q), (¬q↔¬p)↔¬(¬p↔q), (¬q↔¬p)↔¬(¬q↔p), (¬q↔¬p)↔¬(q↔¬p).

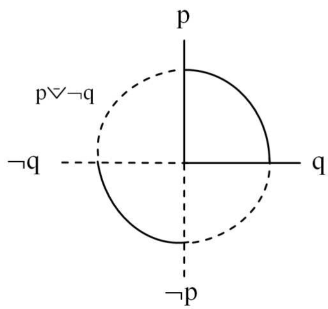
If we move the negative sign of the right side of each above equivalence to the left, the new formed equivalence will still be effective. For instance, from '(p↔q)↔¬(p↔¬q)', we can get '¬(p↔q)↔(p↔¬q)'.

(4) We can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'Vb' from 'Va', or get 'Va' from 'Vb'.

3.6. Incompatible Disjunctive Graphs



Graph 13(a)



Graph 13(b)

In graph VIa, the circumference stands for truth-graph 'p∇q', 'q∇p', '¬p∇¬q', or '¬q∇¬p'; in graph VIb, the circumference stands for truth-graph 'p∇¬q', '¬q∇p', '¬p∇q', or 'q∇¬p'.

We can see the relation between VIa & VIb would now appear as follows: VIa=¬VIb or ¬VIa=VIb.

This means:

(1) The equivalent expressions 'p∇q', 'q∇p', '¬p∇¬q', and '¬q∇¬p' are all equivalent, and can be represented by the same truth-graph;

(2) The equivalent expressions 'p∇¬q', '¬q∇p', '¬p∇q', and 'q∇¬p' are all equivalent, and can be represented by the same truth-graph;

(3) We have these equivalent expressions:
(p∇q)↔¬(p∇¬q), (p∇q)↔¬(¬q∇p), (p∇q)↔¬(¬p∇q), (p∇q)↔¬(q∇¬p), (q∇p)↔¬(p∇¬q), (q∇p)↔¬(¬q∇p), (q∇p)↔¬(¬p∇q), (q∇p)↔¬(q∇¬p), (¬p∇¬q)↔¬(p∇¬q), (¬p∇¬q)↔¬(¬q∇p), (¬p∇¬q)↔¬(¬p∇q), (¬p∇¬q)↔¬(q∇¬p), (¬q∇¬p)↔¬(p∇¬q), (¬q∇¬p)↔¬(¬q∇p), (¬q∇¬p)↔¬(¬p∇q), (¬q∇¬p)↔¬(q∇¬p).

If we move the negative sign of the right side of each above equivalence to the left, the new formed equivalence will still be effective. For instance, from '(p∇q)↔¬(p∇¬q)', we can get '¬(p∇q)↔(p∇¬q)'.

(4) We can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'VIb' from 'VIa', or get 'VIa' from 'VIb'.

Contrast VI with V, the relation among them would now appear as follows:

Va=VIb.

This means:

(1) We can get these equivalent expressions:
$(p\leftrightarrow q)\leftrightarrow(p\triangledown\neg q)$, $(p\leftrightarrow q)\leftrightarrow(\neg q\triangledown p)$, $(p\leftrightarrow q)\leftrightarrow(\neg p\triangledown q)$, $(p\leftrightarrow q)\leftrightarrow(q\triangledown\neg p)$, $(q\leftrightarrow p)\leftrightarrow(p\triangledown\neg q)$, $(q\leftrightarrow p)\leftrightarrow(\neg q\triangledown p)$, $(q\leftrightarrow p)\leftrightarrow(\neg p\triangledown q)$, $(q\leftrightarrow p)\leftrightarrow(q\triangledown\neg p)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow(p\triangledown\neg q)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow(\neg q\triangledown p)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow(\neg p\triangledown q)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow(q\triangledown\neg p)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow(p\triangledown\neg q)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow(\neg q\triangledown p)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow(\neg p\triangledown q)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow(q\triangledown\neg p)$.

(2) 'Va' and 'VIb' share the same truth-graph.

Vb=VIa.

This means:

(1) We can get these equivalent expressions:
$(p\leftrightarrow\neg q)\leftrightarrow(p\triangledown q)$, $(p\leftrightarrow\neg q)\leftrightarrow(q\triangledown p)$, $(p\leftrightarrow\neg q)\leftrightarrow(\neg p\triangledown\neg q)$, $(p\leftrightarrow\neg q)\leftrightarrow(\neg q\triangledown\neg p)$, $(\neg p\leftrightarrow q)\leftrightarrow(p\triangledown q)$, $(\neg p\leftrightarrow q)\leftrightarrow(q\triangledown p)$, $(\neg p\leftrightarrow q)\leftrightarrow(\neg p\triangledown\neg q)$, $(\neg p\leftrightarrow q)\leftrightarrow(\neg q\triangledown\neg p)$, $(\neg q\leftrightarrow p)\leftrightarrow(p\triangledown q)$, $(\neg q\leftrightarrow p)\leftrightarrow(q\triangledown p)$, $(\neg q\leftrightarrow p)\leftrightarrow(\neg p\triangledown\neg q)$, $(\neg q\leftrightarrow p)\leftrightarrow(\neg q\triangledown\neg p)$, $(q\leftrightarrow\neg p)\leftrightarrow(p\triangledown q)$, $(q\leftrightarrow\neg p)\leftrightarrow(q\triangledown p)$, $(q\leftrightarrow\neg p)\leftrightarrow(\neg p\triangledown\neg q)$, $(q\leftrightarrow\neg p)\leftrightarrow(\neg q\triangledown\neg p)$.

(2) 'Vb' and 'VIa' share the same truth-graph.

Va=¬VIa or ¬Va=VIa.

This means:

(1) The equivalent expressions '$p\leftrightarrow q$', '$q\leftrightarrow p$', '$\neg p\leftrightarrow\neg q$', and '$\neg q\leftrightarrow\neg p$' are all equivalent, and can be represented by the same truth-graph;

(2) The equivalent expressions '$p\triangledown q$', '$q\triangledown p$', '$\neg p\triangledown\neg q$', and '$\neg q\triangledown\neg p$' are all equivalent, and can be represented by the same truth–graph;

(3) We have these equivalent expressions:
$(p\leftrightarrow q)\leftrightarrow\neg(p\triangledown q)$, $(p\leftrightarrow q)\leftrightarrow\neg(q\triangledown p)$, $(p\leftrightarrow q)\leftrightarrow\neg(\neg p\triangledown\neg q)$, $(p\leftrightarrow q)\leftrightarrow\neg(\neg q\triangledown\neg p)$, $(q\leftrightarrow p)\leftrightarrow\neg(p\triangledown q)$, $(q\leftrightarrow p)\leftrightarrow\neg(q\triangledown p)$, $(q\leftrightarrow p)\leftrightarrow\neg(\neg p\triangledown\neg q)$, $(q\leftrightarrow p)\leftrightarrow\neg(\neg q\triangledown\neg p)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow\neg(p\triangledown q)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow\neg(q\triangledown p)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow\neg(\neg p\triangledown\neg q)$, $(\neg p\leftrightarrow\neg q)\leftrightarrow\neg(\neg q\triangledown\neg p)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow\neg(p\triangledown q)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow\neg(q\triangledown p)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow\neg(\neg p\triangledown\neg q)$, $(\neg q\leftrightarrow\neg p)\leftrightarrow\neg(\neg q\triangledown\neg p)$.

If we move the negative sign of the right side of each above equivalence to the left, the new formed equivalence will still be effective. For instance, from '$(p\leftrightarrow q)\leftrightarrow\neg(p\triangledown q)$', we can get '$\neg(p\leftrightarrow q)\leftrightarrow(p\triangledown q)$'.

(4) We can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'VIa' from 'Va', or get 'Va' from 'VIa'.

Vb=¬VIb or ¬Vb=VIb.

This means:

(1) The equivalent expressions 'p↔¬q', '¬p↔q', '¬q↔p', and 'q↔¬p' are all equivalent, and can be represented by the same truth-graph;

(2) The equivalent expressions 'p∇¬q', '¬q∇p', '¬p∇q', and 'q∇¬p' are all equivalent, and can be represented by the same truth-graph;

(3) We have these equivalent expressions:

(p↔¬q)↔¬(p∇¬q), (p↔¬q)↔¬(¬q∇p), (p↔¬q)↔¬(¬p∇q), (p↔¬q)↔¬(q∇¬p), (¬p↔q)↔¬(p∇¬q), (¬p↔q)↔¬(¬q∇p), (¬p↔q)↔¬(¬p∇q), (¬p↔q)↔¬(q∇¬p), (¬q↔p)↔¬(p∇¬q), (¬q↔p)↔¬(¬q∇p), (¬q↔p)↔¬(¬p∇q), (¬q↔p)↔¬(q∇¬p), (q↔¬p)↔¬(p∇¬q), (q↔¬p)↔¬(¬q∇p), (q↔¬p)↔¬(¬p∇q), (q↔¬p)↔¬(q∇¬p).

If we move the negative sign of the right side of each above equivalence to the left, the new formed equivalence will still be effective. For instance, from '(p↔¬q)↔¬(p∇¬q)', we can get '¬(p↔¬q)↔(p∇¬q)'.

(4) We can, by changing solid lines to dotted lines, and dotted lines to solid lines, get 'VIb' from 'Vb', or get 'Vb' from 'VIb'.

In summary, we have these relations within a type of or among different types of basic truth-graphs:

Ia=¬Ic; ¬Ia=Ic; Ib=¬Id; ¬Ib=Id.

IIa=¬IIIc or ¬IIa=IIIc; IIb=¬IIId or ¬IIb=IIId.

IIc=¬IIIa or ¬IIc=IIIa; IId=¬IIIb or ¬IId=IIIb.

IIa=¬IVb or ¬IIa=IVb; IIb=¬IVa or ¬IIb=IVa.

IIc=¬IVd or ¬IIc=IVd; IId=¬IVc or ¬IId=IVc.

IIIa=IVd; IIIb=IVc; IIIc=IVb; IIId=IVa.

Va=¬Vb or ¬Va=Vb; VIa=¬VIb or ¬VIa=VIb.

Va=VIb; Vb=VIa; Va=¬VIa or ¬Va=VIa; Vb=¬VIb or ¬Vb=VIb.

The negative graphs above can reflect every kind of truth combination of propositional variables 'p' and 'q', and every other type of truth-graph stands for two or four of the simplest truth expressions, each of which contains only one connective word of truth value and only one propositional variable in its sub-proposition. Therefore, I name these types of truth-graph 'basic truth-graph' or 'fundamental truth-graph.' According to logical connectives, the basic truth-graphs are divided into the six types, and there are different cases or sorts in each type; but according to the graphs, the basic truth-graphs can be divided into fourteen different cases or sorts. Sometimes, we can easily determine the truth value of a complicated compound proposition based upon basic or fundamental truth-graphs and basic truth-tables.

4. Application of the Truth-Graph Method

In some cases, it is much easier to distinguish, by combining truth-graph analysis with simple truth-table analysis, among tautologous, contradictory, or consistent expressions than to do so, by using only the method of a truth-table. Let us first discuss, in an identical coordinate, how to determine the type of graph by two other truth-graphs. I have already stated that a circumference in a truth-graph stands for the truth value of a proposition, which may be a tautologous graph, a contradictory graph or a consistent graph. We can, by solid line and dotted line, distinguish these three kinds of truth-graphs. The tautologous graph is shown by the whole circumference made of the solid line, the contradictory graph is shown by the whole circumference made of the dotted line, and the consistent graph is shown by the circumference made of solid and dotted lines.

A circumference in a coordinate (by 'p-¬p, ¬q-q') consists of four arcs, that is, the arc between 'p' and 'q' (by 'Apq'), the arc between '¬p' and 'q' (by 'A¬pq'), the arc between '¬p' and '¬q' (by 'A¬p¬q'), and the arc between 'p' and '¬q' (by 'Ap¬q'). For instance, circumference 1 consists of Apq(1), A¬pq(1), A¬p¬q(1) and Ap¬q(1), circumference 2 consists of Apq(2), A¬pq(2), A¬p¬q(2) and Ap¬q(2); and so on. In an identical coordinate we can express the relations of circumference 1 and circumference 2, i.e., we can easily evaluate the truth value of a complicated compound proposition. Here are some examples.

Example 1. Determine by truth-graph analysis whether an equivalent expression '¬(p∨q)↔(¬p∧¬q)' is right.

Step 1: Draw out the basic truth-graphs of p∨q (graph14, circumference1) in an identical coordinate (graph 14, p-¬p, ¬q-q).

Step2: According to circumference 1, draw out the truth-graph of ¬(p∨q) (graph 14, circumference 2) in the identical truth coordinate. Its algorithm is:

Apq(1)=T, so Apq(2)=¬Apq(1)=¬T=F;

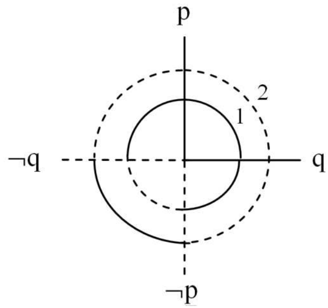
A¬pq(1)=T, so A¬pq(2)=¬A¬pq(1)=¬T=F;

A¬p¬q(1)=F, so A¬p¬q(2)=¬A¬p¬q(1)=¬F=T;

Ap¬q(1)=T, so Ap¬q(2)=¬Ap¬q(1)=¬T=F.

Thus, circumference 2=Apq(2)+A¬pq(2) +¬p¬q(2)+Ap¬q(2) =F+F+T +F.

Step3: Find out whether the truth-graph of ¬(p∨q) is identical with the basic truth-graph of ¬p∧¬q. We can see the two graphs are identical, so the equivalent expression '¬(p∨q) ↔(¬p∧¬q)' is right (graph 14).



Graph 14

Example 2. Determine by truth-graph analysis the truth value of '(p∨q)∨(p↔q)'.

Step 1: Draw out the truth-graphs of p∨q (graph 15, circumference 1) and p↔q (graph 15, circumference 2) separately in an identical coordinate (graph 15, p-¬p, ¬q-q).

Step 2: According to circumference 1 and circumference 2, draw out the truth-graph of (p∨q)∨(p↔q) (graph 15, circumference 3) in the identical truth coordinate. Its algorithm is:

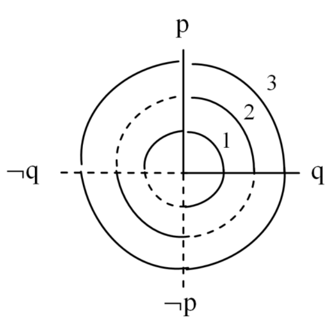
Apq(1)=T, Apq(2)=T, so Apq(3)=Apq(1) ∨Apq(2)=T∨T=T;

A¬pq(1)=T, A¬pq(2)=F, so A¬pq(3)=A¬pq(1)∨A¬pq(2)=T∨F=T;

A¬p¬q(1)=F, A¬p¬q(2)=T, so A¬p¬q(3)=A¬p¬q(1)∨A¬p¬q(2)=F∨T=T;

Ap¬q(1)=T, Ap¬q(2)=F, so Ap¬q(3)=Ap¬q(1)∨Ap¬q(2)=T∨F=T;

Thus, circumference 3=Apq(3)+A¬pq(3)+¬p¬q(3)+Ap¬q(3)=T+T+T+T=T, and we can see that truth value of circumference 3 is a tautologous expression (graph 15).



Graph 15

Example 3. Determine by truth-graph analysis the truth value of '(¬p∧¬q) →(p↔¬q)'.

Step 1: Draw out the truth-graphs of ¬p∧¬q (graph 16, circumference 1) and p↔¬q (graph 16, circumference 2) separately in an identical coordinate (graph 16, p-¬p, ¬q-q).

Step 2: According to circumference 1 and circumference 2, draw out the truth-graph of (¬p∧¬q) →(p↔¬q) (graph 16, circumference 3) in the identical truth coordinate. Its algorithm is:

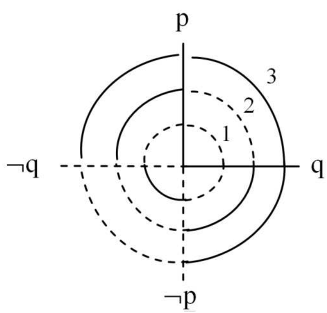
Apq(1)=F, Apq(2)=F, so Apq(3)=Apq(1)→Apq(2)=F→F=T;

A¬pq(1)=F, A¬pq(2)=T, so A¬pq(3)=A¬pq(1)→A¬pq(2)=F→T=T;

A¬p¬q(1)=T, A¬p¬q(2)=F, so A¬p¬q(3)=A¬p¬q(1)→A¬p¬q(2)= T→F=F;

Ap¬q(1)=F, Ap¬q(2)=T, so Ap¬q(3)=Ap¬q(1) →Ap¬q(2)=F→T=T;

Thus, circumference 3=Apq(3)+A¬pq(3)+¬p¬q(3)+Ap¬q(3) =T+T+F +T, and we can see that truth value of circumference 3 is a consistent graph, and formula '(¬p∧¬q) →(p↔¬q)' is a consistent expression (graph 16).



Graph 16

Example 4. Determine by truth-graph analysis the truth value of '(¬p∨ (p→q)) ∧(p∧¬q)'.

Step 1: Draw out the truth-graphs of ¬p (graph 17, circumference 1), p→q (graph 17, circumference 2), and p∧¬q (graph 17, circumference 3) separately in an identical coordinate (graph 17, p-¬p, ¬q-q).

Step 2: According to circumference 1 and circumference 2, draw out the truth-graph of ¬p∨(p→q) (graph 17, circumference 4) in the identical truth coordinate. Its algorithm is:

Apq(1)=F, Apq(2)=T, so Apq(4)=Apq(1)∨Apq(2)=T∨T=T;

A¬pq(1)=T, A¬pq(2)=T, so A¬pq(4)=A¬pq(1)∨A¬pq(2)=T∨T=T;

A¬p¬q(1)=T, A¬p¬q(2)=T, so A¬p¬q(4)=A¬p¬q(1) ∨A¬p¬q(2)= T∨T=T;

Ap¬q(1)=F, Ap¬q(2)=F, so Ap¬q(4)=Ap¬q(1)∨Ap¬q(2)=F∨F=F.

Step 3: According to circumference 4 and circumference 3, draw out the truth-graph of (¬p∨(p→q)) ∧(p∧¬q) (graph 17, circumference 5) in the identical truth coordinate.

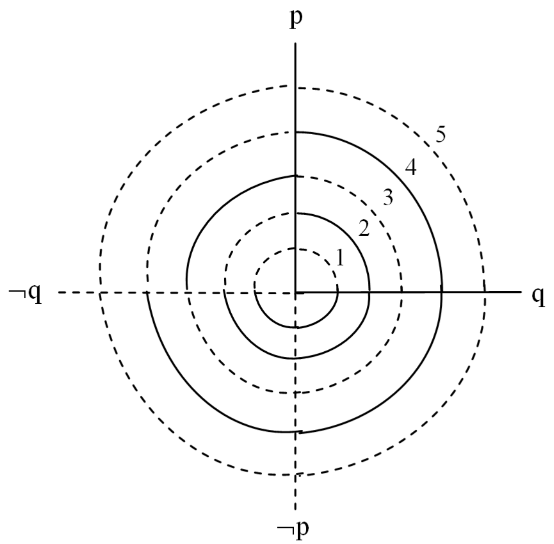
Apq(4)=T, Apq(3)=F, so Apq(5)=Apq(4)∧Apq(3)=T∧F=F;

A¬pq(4)=T, A¬pq(3)=F, so A¬pq(5)=A¬pq(4)∧A¬pq(3)=T∧F=F;

A¬p¬q(4)=T, A¬p¬q(3)=F, so A¬p¬q(5)=A¬p¬q(4)∧A¬p¬q(3)= T∧F=F;

Ap¬q(4)=F, Ap¬q(3)=T, so Ap¬q(5)=Ap¬q(4)∧Ap¬q(3)=F∧T=F.

Thus, circumference 5=Apq(5)+A¬pq(5)+¬p¬q(5)+Ap¬q(5)=F+F+F +F=F, we can see that circumference 5 is contradictory graph, and formula '(¬p∨(p→q)) ∧(p∧¬q)' is contradictory expression (graph 17).



Graph 17

Example 5. Find out by truth-graph analysis the simplest equivalent expression of '(p↔q)∧(p→¬q) →(¬p∇¬q)'.

Step 1: Draw out the truth-graphs of p↔q (graph 18, circumference 1, p→¬q (graph 18, circumference 2), and ¬p∇¬q (graph 18, circumference 3) separately in an identical coordinate (graph 18, p-¬p, ¬q-q).

Step 2: According to circumference 1 and circumference 2, draw out the truth-graph of (p↔q)∧(p→¬q) (graph 18, circumference 4) in the identical truth coordinate. Its algorithm is:

Apq(1)=T, Apq(2)=F, so Apq(4)=Apq(1)∧Apq(2)=T∧F=F;

A¬pq(1)=F, A¬pq(2)=T, so A¬pq(4)=A¬pq(1)∧A¬pq(2)=F∧T=F;

A¬p¬q(1)=T, A¬p¬q(2)=T, so A¬p¬q(4)=A¬p¬q(1)∧A¬p¬q(2)= T∧T=T;

Ap¬q(1)=F, Ap¬q(2)=T, so Ap¬q(4)=Ap¬q(1)∧Ap¬q(2)=F∧T=F.

Thus, circumference 4=Apq(4)+A¬pq(4)+¬p¬q(4)+Ap¬q(4)=F+F+T +F.

Step 3: According to circumference 4 and circumference 3, draw out the truth-graph of (p↔q)∧(p→¬q) →(¬p∇¬q) (graph 18, circumference 5) in the identical truth coordinate. Its algorithm is:

Apq(4)=F, Apq(3)=F, so Apq(5)=Apq(4) →Apq(3)=F→F=T;

A¬pq(4)=F, A¬pq(3)=T, so A¬pq(5)=A¬pq(4) →A¬pq(3) =F→T =T;

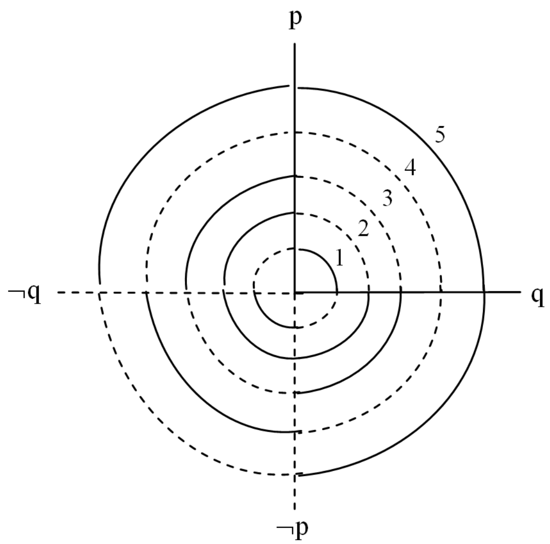
A¬p¬q(4)=T, A¬p¬q(3)=F, so A¬p¬q(5)=A¬p¬q(4) →A¬p¬q(3)= T→F=F;

Ap¬q(4)=F, Ap¬q(3)=T, so Ap¬q(5)=Ap¬q(4) →Ap¬q(3)=F→T=T.

Step 4: Determine which type of basic truth-graph is identical with the truth-graph drawing in the third step and write out the corresponding simplest truth expression.

Thus, circumference 5=Apq(5)+A¬pq(5)+¬p¬q(5)+Ap¬q(5)=T+T+F +T, we can see that circumference 5 is the consistent graph, and the primitive formula is the consistent expression.

From the graph, we can see that circumference 5 is identical with the basic truth-graph of p∨q, ¬p→q, or ¬q→p (graph 18).



Graph 18

The method of truth-graph expounded above only involves two proposition variables. If three propositional variables are touched upon, the graph is slightly more complicated. We should first convert the expression with more than two variables into that only with two variables. Take an example: determine by truth-graph analysis whether an equivalent expression '¬(r∨s∨q)↔(¬(r∨s)∧¬q)' is right. There are three propositional variables in the expression, so, in order to use the method of truth-graph, we should add a replacement step first, i.e., to replace 'r∨s' with 'p'. Thus we just need to determine the truth value of expression '¬(p∨q)↔(¬p∧¬q)'.

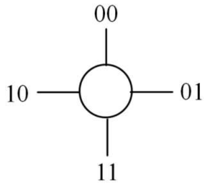
5. Truth-Graph and Leśniewski's Notation

Leśniewski invented a kind of idiosyncratic notation for one- and two-place propositional connections. His notation seems to have been used for the first time in (Leśniewski 1929), and explains it in his (1938).[1] The most important feature of the notation is that the shape of a symbol encodes its truth-table. In contrast, a truth-graph which is made up of a truth-value coordinate and four arcs displays its truth-table almost entirely through its image. In this section I will describe Leśniewski's notion and compare it with my truth-graph, and later on I will show a few examples to reveal the merits of the two different methods that are generated by distinct signs.

Leśniewski's notation is formally a prefix notation. Functors are placed before arguments that are put in parentheses and are not separated from each other by any punctuation. Universal quantification binds variables of any semantic category, and the whole sequence of those variables is placed within the lower corner-brackets: ⌊ ⌋. The scope of a quantifier is put in the upper-corner brackets: ⌈ ⌉. For example, a universal quantifier binding propositional variables p_1, p_2 is: ⌊p_1 $p_{2,}$⌋ What we often write as '∀pp' is expressed as '⌊p⌋ ⌈p⌉'. Compared with notation, the truth graph is not a kind of notation, but just a convenient graphic method to help determine the truth value of a sentence and to leave aside the problem of quantification.

5.1 Leśniewski's 2-Place Connectives

Leśniewski's 2-place functor consists of two parts: a hub and four possible spokes. The hub is marked by a circle: 'o', and the spokes by four possible lines which can be added in four directions: up, down, left, and right. The different directions express a type of truth-value combination of two arguments (e.g. argument 'p' is true or has value 1, and 'q' is false or has value 0), and the line itself shows that the whole expression is true. The schema is as follows:



Schema 1

Schema 1 shows the four possible lines: a line to the left which denotes that the arguments have values respectively 1, 0, and the whole has value 1; a line to the right which denotes that the arguments have values respectively 0, 1, and the whole has value 1; a line upwards which denotes that the arguments have values 0, 0, and the whole has value 1; a line downwards which denotes that the arguments have values 1, 1, and the whole has value 1. Thus, we have 16 possible 2-place functors, each of which is characterized by a four-place sequence of elements 1 and 0. The whole expression is true when arguments have values, respectively, 11, 00, 10, 01. The 16 functors and their names as well as characteristics are as follows:[2]

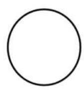
Functor	Name	Characteristics
	2-place falsum	0000
	Conjunction	1000
	Conegation	0001
	Distinction	0100
	Contradistinction	0010
	Equivalence	1001
	Disjunction	0110
	Antecedent Affirmation	1100
	Consequent Affirmation	1010
	Antecedent Negation	0011
	Consequent Negation	0101
	Exclusion	0111
	Implication	1011
	Counterimplication	1101
	Alternation	1110
	2-place verum	1111

A common feature of the notation and the truth-graph is that there is a close correspondence between geometrical properties and logical properties, and the notation can be a part of a formula. A truth-graph, however, is just a graphic that shows all its logical characteristics directly only by geometrical properties. As a method, a truth-graph is not used in a formula, but just to help evaluate the truth-value of a formula.

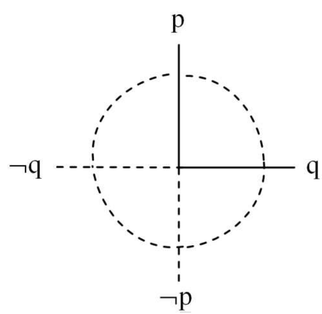
5.2. Correspondence between Notation and Truth-Graph

From the perspective of geometrical or logical properties, there is a correspondence between Leśniewski's notation and my truth-graph. Leśniewski's 16 possible 2-place notations can all be shown by the truth-graph. Now let us list and display respectively which notation is associated with which truth-graph.

1. 2-place falsum and its corresponding truth-graph



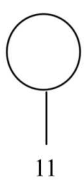
Schema 2



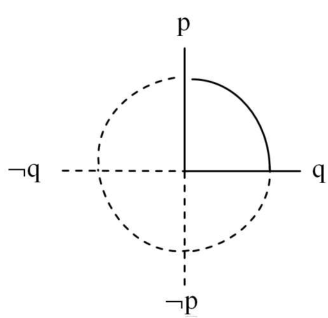
Graph 19

Schema 2 stands for a two-place basic formula: $\lfloor p\ q \rfloor \circ (pq)$. Graph 19 stands for an assertion: $\vdash p \wedge \neg p$ (It is asserted that $p \wedge \neg p$), or $\vdash q \wedge \neg q$. Both schema 2 and graph 19 stand respectively for a contradiction; i.e., the whole expression is eternally false whether 'p' or 'q' is true (has value 1) or not.

2. Conjunction and its corresponding truth-graph



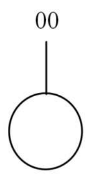
Schema 3



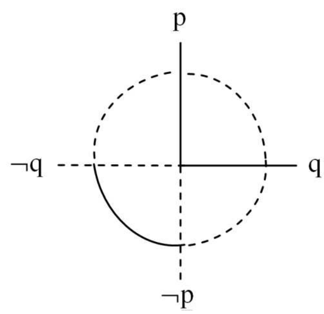
Graph 20

Schema 3 stands for a two-place basic formula: $\lfloor p\ q \rfloor$⫯(pq), while graph 20 (also see graph 9 (a)) stands for a basic assertion: $\vdash p \wedge q$. Both schema 3 and graph 20 show that an expression is true iff all its arguments are true (have value 1).

3. Conegation and its corresponding truth-graph



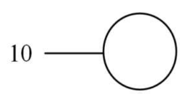
Schema 4



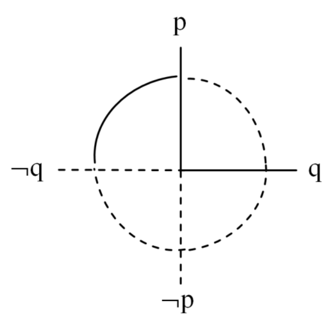
Graph 21

Schema 4 stands for a two-place basic formula: $\lfloor p\ q \rfloor$⫰(pq) while graph 21 (also see graph 9 (c)) stands for an assertion: $\vdash \neg p \wedge \neg q$. Schema 4 and graph 21 all show that an expression is true iff all its arguments are false (have value 0).

4. Distinction and its corresponding truth-graph



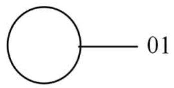
Schema 5



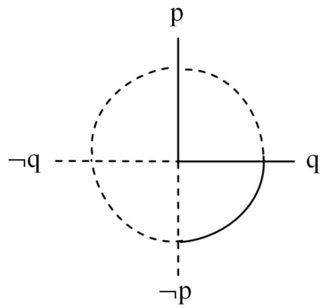
Graph 22

Schema 5 stands for a two-place basic formula: $\lfloor p\ q \rfloor$⊸(pq), while graph 22 (also see graph 9 (b)) stands for an assertion: $\vdash p \wedge \neg q$. Schema 5 and graph 22 all show that an expression is true iff 'p' is true (has value 1) and 'q' is false (has value 0).

5. Contradistinction and its corresponding truth-graph



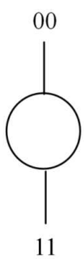
Schema 6



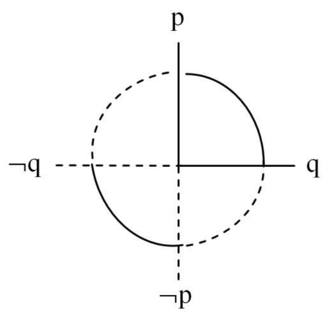
Graph 23

Schema 6 stands for a two-place basic formula: ⌊p q⌋⟜(pq), while graph 23 (also see graph 9 (d)) stands for an assertion: ⊢¬p∧q. Schema 6 and graph 23 all show that an expression is true iff ‘p’ is false (has value 0) and ‘q’ is true (has value 1).

6. Equivalence and its corresponding truth-graph



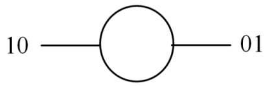
Schema 7



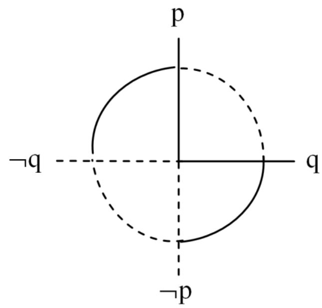
Graph 24

Schema 7 stands for a two-place basic formula: ⌊p q⌋⫯(pq), while graph 24 (also see graph 12 (a)) stands for an assertion: ⊢p↔q. Schema 6 and graph 24 all show that an expression is true iff both ‘p’ and ‘q’ is true (have value 1), or both ‘p’ and ‘q’ is false (have value 0).

7. Disjunction and its corresponding truth-graph



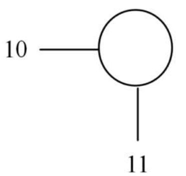
Schema 8



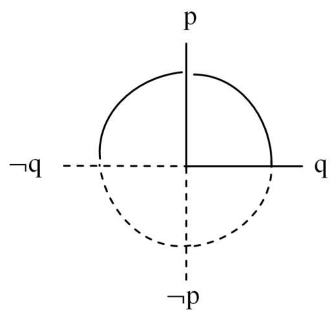
Graph 25

Schema 8 stands for a two-place basic formula: ⌊p q⌋-o-(pq), while graph 25 (also see graph 13 (a)) stands for an assertion: ⊢ p∇q. In short, Schema 8 and graph 25 all show that an expression is true iff 'p' is true (has value 1) and 'q' is false (has value 0), or 'p' is false and 'q' is true.

8. Antecedent Affirmation and its corresponding truth-graph



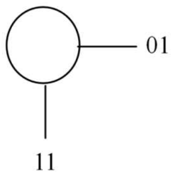
Schema 9



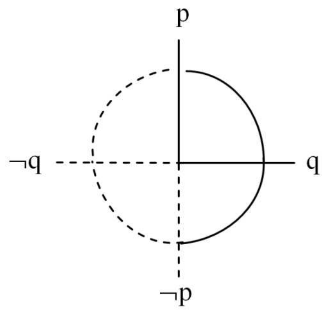
Graph 26

Schema 9 stands for a two-place basic formula: ⌊p q⌋-ọ(pq), which means that 'p' is true (has value 1) and 'q' is false (has value 0), or both 'p' and 'q' are true. Graph 26 (also see graph 8 (a), 8(b)) stands for an assertion: ⊢ p. It means that 'p' is true whether 'q' is true or not, or that '¬p' is false whether 'q' is true or not.

9. Consequent Affirmation and its corresponding truth-graph



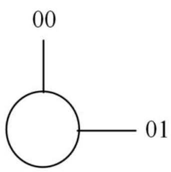
Schema 10



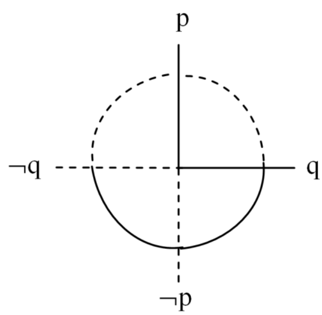
Graph 27

Schema 10 stands for a two-place basic formula: $\lfloor p\ q \rfloor$⚲(pq), which means the whole expression is true iff ‘p’ is false (has value 0) and ‘q’ is true (has value 1), or both ‘p’ and ‘q’ are true. Graph 27 (also see graph 8 (a), 8(d)) stands for an assertion: $\vdash q$. It means that ‘q’ is true whether ‘p’ is true or not, or that ‘¬q’ is false whether ‘q’ is true or not.

10. Antecedent Negation and its corresponding truth-graph



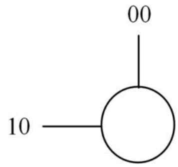
Schema 11



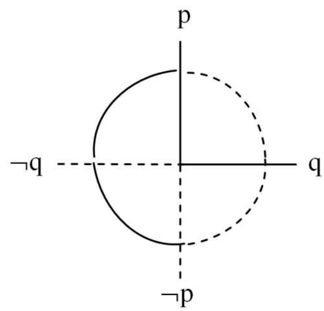
Graph 28

Schema 11 stands for a two-place basic formula: $\lfloor p\ q \rfloor$⚲(pq), which means the whole expression is true iff ‘p’ is false (has value 0) and ‘q’ is true (has value 1), or both ‘p’ and ‘q’ are false. Graph 28 (also see graph 8 (c), 8(d)) stands for an assertion: $\vdash \neg p$. It means that ‘p’ is false whether ‘q’ is true (has value 1) or not, or that ‘¬p’ is true whether ‘q’ is true or not.

11. Consequent Negation and its corresponding truth-graph



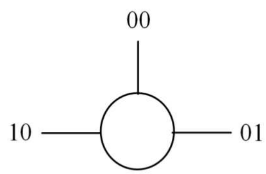
Schema 12



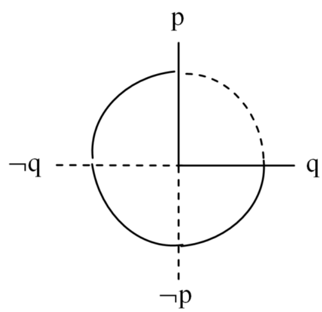
Graph 29

Schema 12 stands for a two-place basic formula: ⌊p q⌋-ȯ(pq), which means the whole expression is true iff 'p' is true (has value 1) and 'q' is false (has value 0), or both 'p' and 'q' are false. Graph 29 (also see graph 8(b), 8(c)) stands for an assertion: ⊢¬q. It means that 'q' is false (has value 0) whether 'p' is true or not, or that '¬q' is true whether 'p' is true or not.

12. Exclusion and its corresponding truth-graph



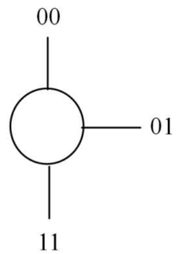
Schema 13



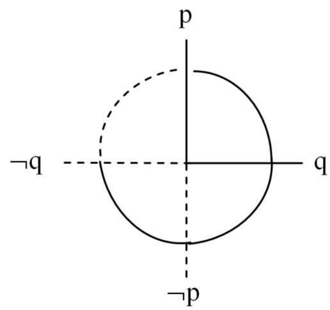
Graph 30

Schema 13 stands for a two-place basic formula: ⌊p q⌋-ȯ-(pq), while graph 30 (also see graph 11(b)) stands for an assertion: ⊢p→¬q. Both schema 13 and graph 30 mean that the whole expression is true iff 'p' is true (has value 1) and 'q' is false (has value 0), or, 'p' is false and 'q' is true, or, both 'p' and 'q' are false.

13. Implication and its corresponding truth-graph



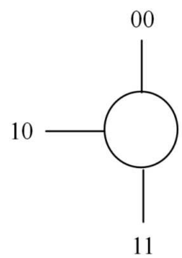
Schema 14



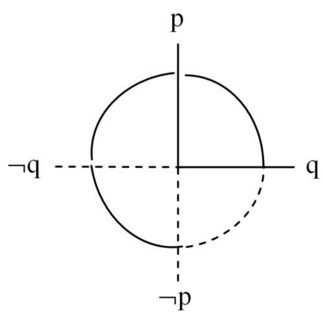
Graph 31

Schema 14 stands for a two-place basic formula: ⌊p q⌋ϕ-(pq), while graph 31 (also see graph 10(d), 11(a)) stands for assertions: ⊢ ¬p∨q, ⊢ p→q. Both schema 14 and graph 31 mean that the whole expression is true iff 'p' is false (has value 0) and 'q' is true (has value 1), or, both 'p' and 'q' are true, or, both 'p' and 'q' are false.

14. Counterimplication and its corresponding truth-graph



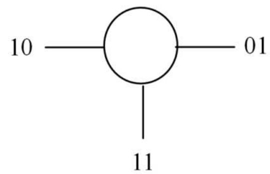
Schema 15



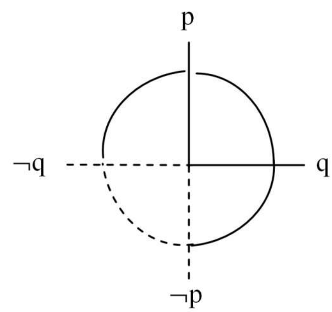
Graph 32

Schema 15 stands for a two-place basic formula: ⌊p q⌋-ϕ(pq), while graph 32 (also see graph 10(b), 11(c)) stands for assertions: ⊢ p∨¬q, ⊢ ¬p→¬q. Both schema 15 and graph 32 mean that the whole expression is true iff 'p' is true (has value 0) and 'q' is false (has value 0), or, both 'p' and 'q' are true, or, both 'p' and 'q' are false.

15. Alternation and its corresponding truth-graph



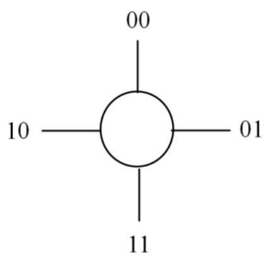
Schema 16



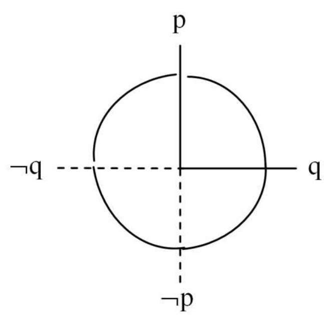
Graph 33

Schema 16 stands for a two-place basic formula: ⌊p q⌋-⫯-(pq), while graph 33 (also see graph 10(a), 11(d)) stands for assertions: ⊢ p∨q, ⊢ ¬p→q. Both schema 16 and graph 33 mean that the whole expression is true iff 'p' is true (has value 0) and 'q' is false (has value 0), or, 'p' is false and 'q' is true, or, both 'p' and 'q' are true.

16. 2-place verum and its corresponding truth-graph



Schema 17



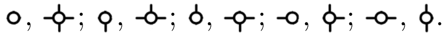
Graph 34

Schema 17 stands for a two-place basic formula: ⌊p q⌋-⌖-(pq). Graph 34 stands for an assertion: ⊢ p∨¬p, or ⊢ q∨¬q. Both schema 17 and graph 34 stand respectively for a tautology, i.e., the whole expression is eternally true whether 'p' or 'q' is true or not.

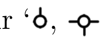
5.3. Relationship between Functors

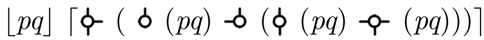
In my opinion, there are at least four types of relations among 2-place functors, i.e., contradiction, contrariety, subcontrariety, and subalternation. These relations can be visually understood from Leśniewski's schema and my truth-graph. According to these relations, we may find some fundamental inference formulae.

Contradiction. Two 2-place functors, say *f*, *g* are contradictory of each other iff where there is a line on *f*, there is no line on *g*, and where there is no line on *f*, there is a line on *g*. For instance, the following pairs are contradictory of each other:



The contradictory correlation between schemata may also be visually shown by corresponding graph, where all solid arcs in a graph will become the dotted in another graph, and all dotted arcs in a graph will become the solid in the other one.

From the perspective of logic, contradiction means that one proposition is true, while the other is false, and one is false, while the other is true. The contradictory correlation between shapes can help to write valid inferential formulae. For example, pair '' means following right expression:

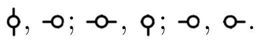


In this sentential connectives, p, q are propositional variables. In modern notation:

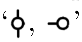
$$\neg p \wedge \neg q \rightarrow \neg(p \vee q)$$

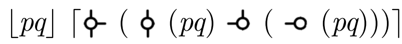
which may be easily validated by truth-graph method.

Contrariety. Two 2-place functors, say *f*, *g* are contrary of each other iff where there is a line on *f*, there will be no line on *g*, but where there is no line on *f*, there might be no line on *g*. For instance, the following pairs are contrary of each other:

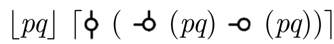


Similarly, the contrary correlation between schemata can also be visually reflected by corresponding graph, where all solid arcs in a graph will become the dotted in another graph, but all dotted arcs in a graph might still be the dotted in the other one.

Logically, the contrary correlation between shapes can help to write valid compound expression. For example, pair '' implies the following valid expression:



and the following invalid expression:



In modern notation, the valid expression is:

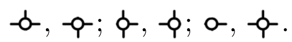
$$(p \leftrightarrow q) \rightarrow \neg(p \wedge \neg q),$$

and the invalid expression is:

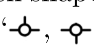
$$\neg q \leftrightarrow (p \wedge \neg q).$$

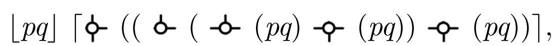
By truth-graph method we can also determine whether the two expressions are true or not.

Subcontrariety. Two 2-place functors, say *f*, *g* are subcontrary of each other iff where there is no line on *f*, there will be a line on *g*, but where there is a line on *f*, there might be a line on *g*. For instance, the following pairs are subcontrary of each other:

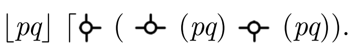


These subcontrary pairs may also be visually shown by corresponding graph, where all dotted arcs in a graph will become the solid in another graph, but all solid arcs in a graph might still be the solid in the other one.

Logically, the subcontrary correlation between shapes signifies valid and invalid compound expression. For instance, pair ‘’ implies the following valid expression:



and the following invalid expression:



In modern notation, the valid expression is:

$$\neg(\neg p \vee \neg q) \rightarrow (p \vee q),$$

and the invalid expression is:

$$(\neg p \vee \neg q) \rightarrow (p \vee q).$$

The truth-value of the whole expression may be easily checked up by truth-graph method.

Subalternation. Two 2-place functors, say *f*, *g* are subalternate of each other iff where there is a line on *f*, there will be a line on *g*, and where there is no line on *g*, there will be no line on *f*. For example, the following pairs are subalternate of each other:

, ; , ; , .

The subalternate pairs may also be visually indicated by corresponding graph, where all solid arcs in graph *j* will still the solid in graph *k*, and all dotted arcs in graph *k* will still be the dotted in graph *j*.

The subalternate correlation between the shapes signifies logically valid and invalid compound expression. For instance, pair ', ' implies the following valid expression:

$$\lfloor pq \rfloor \lceil (\; (\; (pq) \; (pq) \rceil ,$$

and the following invalid expression:

$$\lfloor pq \rfloor \lceil (\; (\; (pq) \; (pq) \rceil ,$$

In modern notation, the valid expression is:

$$(\mathrm{p} \leftrightarrow \neg \mathrm{q}) \rightarrow (\mathrm{p} \rightarrow \neg \mathrm{q}),$$

and the invalid expression is:

$$(\mathrm{p} \rightarrow \neg \mathrm{q}) \rightarrow (\mathrm{p} \leftrightarrow \neg \mathrm{q}).$$

It is not hard to check whether the whole expressions are true or not by using the truth-graph method.

This paper is supported by Chinese National Social Science Fund (11BZX031) in 2011, Chinese Education Ministry Fund for Returned Overseas Professional in 2012, and Scientific Research Fund of Huaqiao University in 2015. I would like to thank Kazimierz Trzęsicki and Yuzhang Zhai for their helpful comments.

NOTES

1 Srzednicki and Stachniak discuss the notation in their (1988), and Luschei also makes some contribution to this notation (Luschei 1962, 289–305).

2 Names of functors are normally stated in the light of Luschei's translation (Luschei 1962: 290–291), with some differences. Numbers indicating the characteristics of a functor are arranged in down-left-right-up order whose present or absent spokes occur around the hub. A present spoke is denoted by '1', which means true, and an absent spoke by '0', which means false.

REFERENCES

Greenstein, C.H. (1978). *Dictionary of logical terms and symbols.* London: Van Nostrand Reinhold Company.

Leśniewski, S. (1929). Grundzüge eines neuen Systems der Grundlagen der Mathematik, §1–11. *Fundamenta Mathematicae*, *14*, 1–81.

Leśniewski, S. (1938). Einleitende Bemerkungen zur Fortsetzung meiner Mitteilung U.D.T. 'Grundzüge eines neuen Systems der Grundlagen der Mathematik'. Widener Library Info Harvard Depository XLL 270.5. Hollis number: 005913328.

Luschei, E. (1962). *The logical systems of Leśniewski.* Amsterdam: North-Holland Press.

Newton-Smith, W.H. (1985). *Logic: An introductory course.* Guernsey, CI: Guernsey Press Company.

Peirce, C.S. (1931–1935). *Collected papers.* C. Hartshorne and P. Weiss, Eds. Cambridge, Mass.: Harvard University Press, Vols. I–VI.

Quine, W.V. (1982). *Method of logic.* Cambridge, Mass.: Harvard University Press.

Srzednicki, J., & Stachniak, Z. (Eds.). (1988). *S. Leśniewski's lecture notes in logic.* Dordrecht: Kluwer Academic.