NOTES FOR SYMBOLIC LOGIC

by
Rudolf Carnap
The University of Chicago

Distributed by
The University of Chicago Bookstore
5802 Ellis Avenue
Chicago, Illinois

NOTES FOR SYMBOLIC LOGIC. FIRST PART

by

R. Carnap

These notes are not meant as an introduction to symbolic logic. They give merely a survey of the symbolism and its rules, with no explanations or only short ones. Sometimes references are given to the following books which supply detailed explanations:

Abbreviation	
Carnap - Abr.	Carnap, Abriss der Logistik. Vienna 1929.
Carnap - Sy.	Carnap, Logical Syntax of Language. N.Y. 1937.
Hilbert	Hilbert and Ackermann, Grundzüge der theoretischen Logik. Berlin 1928.
Lewis	Lewis and Langford, Symbolic Logic. N.Y. 1932.
PM	Whitehead and Russell, Principia Mathematica. Vol. I, Cambridge (1910) 2nd ed. 1925.

1. Use of Letters

Kinds of Symbols	Level	Letters Used: As Constants	Letters Used: As Variables	Designate
sentential symbols	--	A,B,C...	p,q,r...	--(states of affairs)
individual symbols	0	a,b,c...	x,y,z,u...	objects
predicates 1-place	1	P,Q...	F,G...	properties
" 2-place	1	R,S...	H,K...	relations
" (see #9)	2 : n	^2P... ^nP...	F... nF...	propert. of properties
functors (see #10)		k,l...	f,g...	

Examples of Sentences	Translation
P(b)	the object b has the property P; b is P
R(a,b)	the relation R holds between a and b
$^2P(Q)$(see #9)	the property Q has the property (of properties) ^2P; Q is ^2P.

In 'R(a,b)', 'a' and 'b' are called arguments. A predicate is called n-place or of <u>degree</u> n if it requires n arguments. A sentence consisting of an n-place predicate and n arguments is called a <u>full</u> <u>sentence</u> of that predicate.

<u>2. Sentential Calculus</u>

<u>Sentential Connectives</u>.
(Hilbert Kap.I,#1; Carnap-Abr.#3; Carnap-Sy.#5.)

	Explained by Other Connectives	Name	Translation
$\sim A$		negation	not A
$A \vee B$		disjunction	A or B (or both)
$A . B$		conjunction	A and B
$A \supset B$	$\sim A \vee B$	implication	not A, or B; if A then B
$A \equiv B$	$(A . B) \vee (\sim A . \sim B)$ $(A \supset B) . (B \supset A)$	equivalence	A and B, or, not A and not B; if A then B, and if B then A; A if and only if B
$A \mid B$	$\sim(A . B)$ $\sim(A) \vee (\sim B)$	(incompatibility)	not, A and B; either not A or not B or both

'|' is seldom used in practical application.

The truth-value table of a connective states the truth-value -- i.e., truth (T) or falsehood (F) -- of a full sentence with respect to the truth-values of the arguments. (Lewis pp. 200-211; Carnap-Abr. #3; Carnap-Sy.#5)

I.

A	$\sim A$
T	F
F	T

II.

A	B	$A \vee B$	$A . B$	$A \supset B$	$A \equiv B$	$A \mid B$
T	T	T	T	T	T	F
T	F	T	F	F	F	T
F	T	T	F	T	F	T
F	F	F	F	T	T	T

3. Lower Functional Calculus

Universal and Existential Sentences.

(Lewis ch. V; Carnap-Abr. #6; Carnap-Sy. #6)

Examples of Sentences	Kinds of Sentences	Translation
P (x) ↑ free variable	open s.	
bound variable ↓ (x) P(x) ↓ ↓ univ. operator, operand	universal s. closed s.	every object is P
bound var. ↓ (∃x) (P(x)) ↓ ↓ exist. operator, operand	existential s. closed s.	some (at least one) object is P; there is an object which is P

Formative Rules.

An expression is a sentence (of the lower functional calculus, including the sentential calculus) if and only if it has one of the following forms (where at the place of '...' and '---', sentences of any form may stand):

	Form	Kind of Sentence	Examples
1.	a sentential symbol	atomic s.	A; p
2.	a full sentence of a predicate (see #1)	atomic s.	P(a); R(x,y)
3.	an identity sentence	atomic s.	a = b; x = y
4.	~ (…)	molecular s.	~ (A);~ (p ∨ q)
5.	(...) ∨ (---)	molecular s.	(A) ∨ (B)
	(...) . (---)	molecular s.	P(a) . (R(b,x))
	(...) ⊃ (---)	molecular s.	(F(y)) ⊃ (∃x)(F(x))
	(...) ≡ (---)	molecular s.	(p) ≡ (q)

6.	$(x)(\ldots)$ $(\exists x)(\ldots)$	where at the place of 'x' any other indiv. var. may stand	general s.	$(x)((P(x)) \vee (R(x,b)))$ $(\exists x)((y)(R(x,y)))$

Conventions for Avoiding Brackets.

Brackets enclosing a certain expression may be omitted if one of the following conditions is fulfilled:

Rule	The Enclosed Expression Has the Following Form:	The Enclosed Expression Occurs in the Whole Sentence as Follows:
1.	any atomic sentence	as a member of a sentential connection
2a.	negation	as a member of any sent. conn.
2b.	disjunction or conjunction	as a member of an implication or an equivalence
3a.	disjunction	as a member of a disjunction
3b.	conjunction	as a member of a conjunction
	(Rule 3a is justified by Th. 34 (see #7); 3b by Th.35)	
4.	operator with operand	in any way
5.	operand	as the smallest sentence immediately following the operator

Examples:

Rule	Instead of	We May Write
1.	$\sim(A)$ $(P(a)) \vee (R(b,c))$ $(x = y) \supset (y = x)$	$\sim A$ $P(a) \vee R(b,c)$ $x = y \supset y = x$
2a.	$(\sim A) \vee B$ $(\sim A) . B$ $(\sim A) \supset B$ $(\sim A) \equiv B$	$\sim A \vee B$ $\sim A . B$ $\sim A \supset B$ $\sim A \equiv B$
2b.	$(p . q) \supset (p \vee q)$ $(A \vee B) \equiv (C . D)$	$p . q \supset p \vee q$ $A \vee B \equiv C . D$
3a.	$(A \vee B) \vee C$ $A \vee (B \vee C)$	$A \vee B \vee C$

3b.	(A . B) . C A . (B . C .)	A . B . C .
4.	~ ((x) (P(x) ∨ Q(x))) (x) ((y) ((∃z) (T(x,y,z) . A)))	~ (x) (P(x) ∨ Q(x)) (x) (y) (∃z) (T (x,y,z) . A)
5.	(x) (P (x)) ∨ A (∃x) (~ (y) (R (x,y))) . B	(x) P (x) ∨ A (∃x) ~ (y) R (x,y) . B

4. Transformative Rules. (Hilbert, Kap. III, #5; Carnap-Sy. #10)

Primitive Sentences.

1. Sentential Calculus.
   - PS 1. $p \vee p \supset p$
   - PS 2. $p \supset p \vee q$
   - PS 3. $p \vee q \supset q \vee p$
   - PS 4. $(p \supset q) \supset (r \vee p \supset r \vee q)$
2. (Lower) Functional Calculus.
   - PS 5. $(x)F(x) \supset F(y)$
   - PS 6. $F(y) \supset (\exists x)F(x)$
3. Identity.
   - PS 7. $x = x$
   - PS 8. $x = y \supset (F(x) \supset F(y))$

Rules of Inference.

A sentence S_3 is called directly derivable from a sentence S_1 or (in the case of R2) from two sentences S_1 and S_2 -- called the premisses,-- if and only if one of the following conditions is fulfilled:

Rl. Rule of Substitution. S_3 is constructed out of S_1 by substituting for a variable wherever it occurs as a free variable in S_1 a symbol or expression of a suitable kind (see below, (a)-(d)). At all places the same symbol or expression must be substituted. An expression meet not be substituted if it contains a free variable which would be bound, after the substitution, at one of the substitution places.

(a) For a sentential variable any sentence may be substituted.

(b) For an individual variable any individual symbol may be substituted.

(c) For an n-place predicate variable any n-place predicate may be substituted.

(d) For a full sentence S_4 consisting of an n-place predicate variable with n different individual variables as arguments, any sentence may be substituted. In S_1, S_4 is replaced by S_5, and any other full sentence of the same predicate variable occurring in S_1 is replaced by the corresponding sentence constructed out of S_5 by individual substitutions (see example below). An individual variable occurring as a free variable in S_5 but not occurring in S_4 must not be such that it will be a bound variable after the substitution at one of the substitution places.

R2. Rule of Implication. S_1 and S_2 have the forms '...' and '(…) ⊃ (---)' respectively, and S_3 has the form '---', where '...' and '---' stand for two sentences of any form. (In other words, S_2 is an implication sentence with S_1 and S_3 as members.)

R3. Rules of the Operators.

(a) S_1 has the form '(…) ⊃ (---)' where 'x' does not occur in '...' as a free variable. S_3 has the form '(…) ⊃ (x)(---)'.

(b) S_1 has the form '(…) ⊃ (---)' where 'x' does not occur in '---' as a free variable. S_3 has the form (∃x) (…) ⊃ (---)'.

In (a) and (b) '...' and '---' stand for sentences of any form; instead of 'x' any other individual variable may be taken.

Examples :

Rule	One or Two Premisses	Directly Derivable from the Premisses
R1a.	$p \vee \sim p$	$q \vee \sim q$ $A \vee \sim A$ $R(a,b) \vee \sim R(a,b)$
R1b.	$P(x) \vee (\exists y)R(x,y)$	$P(z) \vee (\exists y)R(z,y)$ $P(b) \vee (\exists y)R(b,y)$

R1c.	$F(a) \supset F(b)$	$G(a) \supset G(b)$ $P(a) \supset P(b)$
R1d.	$..F(x)..F(y)..F(a)..F(c)..$	$..P(x) \vee R(x,b)..P(y) \vee R(y,b)..$ $P(a) \vee R(a,b)..\ P(c) \vee R(c,b)..$ (explanation of this below.)
R2.	$A;\ A \supset B$ $P(a);\ P(a) \supset (\exists y)R(a,y)$	B $(\exists y)R(a,y)$
R3a.	$P(a) \supset R(a,x)$	$P(a) \supset (x)R(a,x)$
R3b.	$R(b,y) \supset Q(b)$	$(\exists y)R(b,y) \supset Q(b)$

Explanation for the example for R1d. The premiss is meant as some sentence containing the four full sentences of 'F' given here; the dots indicate the rest of the sentence which is irrelevant to the substitution. Suppose we are to carry out the substitution of '$P(x) \vee R(x,b)$' for '$F(x)$'. '$F(x)$' is replaced by '$P(x) \vee R(x,b)$'; '$F(y)$' is not replaced by the same sentence, but by that sentence which we construct out of it by substituting 'y' for 'x', i.e., the sentence '$P(y) \vee R(y,b)$'; analogously, '$F(a)$' is replaced by '$P(a) \vee R(a,b)$', and '$F(c)$' by '$P(c) \vee R(c,b)$'. Thus the result given above is attained.

5. Definitions. (Carnap-Sy. #8,29.)

A definition is an additional transformative rule which serves for the introduction of a new symbol.

Examples of definitions (the new symbol defined is in (1) to (5) the first symbol of the definiendum., In (6) '≡').

#	Definiendum		Definiens	Kinds of Definitions
1.	5	$=$	$4 + 1$	def. sentence; Explicit def. (in narrow sense)
2.	A	$\equiv$	$R(a,b)$	def. sentence; Explicit def. (in narrow sense)
3.	$Q_1(x)$	$\equiv$	$P(x) \vee R(x,c)$	def. sentence; def. in use
4.	$Q_2(x)$	$\equiv$	$P(z) \vee (\exists y)R(x,y)$	def. sentence; def. in use
5.	'$Q_1(x)$'	for	'$P(x) \vee R(x,c)$'	def. rule; def. in use
6.	'$p \equiv q$'	for	'$(p \supset q) . (q \supset p)$'	def. rule; def. in use

(Sometimes all definitions of these kinds are called explicit -- in the wider sense -- in contradistinction to recursive definitions.)

If a definition-rule of the form " '...' for '---' " is given, it means the following: Whenever a sentence S_2 is constructed out of a sentence S_1 by replacing the expression '...' (not necessarily at all places where it occurs) by the expression '---', then S_2 is directly derivable from S_1 and S_1 from S_2. If free variables occur, then the mutual replacement is permitted for any two expressions constructed out of the definiendum and the definiens by the same substitutions.

In the case of definition-sentences of the forms '… ≡ ---' or '… = ---' analogous mutual replacements can be carried out (by Th. 90 and 91, #8).

Any definition-rule or definition-sentence must fulfill the following conditions: 1. If the definiendum contains free variables, they must be different from one another. 2. The definiens must not contain any free variable not occurring in the definiendum.

Let us take the following as primitive symbols (i.e., undefined symbols) of our language:

(1) logical symbols: '~', '∨', '∃', comma, brackets, all variables.

(2) descriptive symbols: some individual constants and predicate constants (as many as are necessary for the formulation of the theory in question.)

For every defined symbol there must be a chain of definitions ending with the definition of this symbol.This chain must be such that every symbol occurring in a definiens is either one of the primitive symbols or defined by a preceding definition. A defined symbol is called logical if no primitive descriptive symbol occurs in its chain of definitions; otherwise it is called descriptive. A symbol is called indefinite if it is defined by a chain of definitions containing at least one operator; otherwise it is called definite.

Definitions (rules) of some logical symbols.

Def. 1. 'p . q' for '~(~p ∨ ~q)'

Def. 2. 'p⊃ q' for '~p ∨ q'

Def. 3. 'p ≡ q' for '(p ⊃ q) . (q ⊃ p)'

Def. 4. 'p | q' for '~(p . q)'

(In the PS and in R2 (#4), the defined symbol ' ⊃ ' has been used for the sake of brevity. It can easily be eliminated in accordance with Def. 2.)

Instead of '~', ' ∨ ', and ' ∃', we could take '| ' as primitive. In this case we would lay down the following definitions:

Def. A.'~p' for 'p | p'

Def. B.'p ∨ q' for '~p | ~q'

Def. C.'(∃x)F(x)' for '~ (x)~F(x)', (where am other individual variable may be put at the place of 'x').

(For examples of logical definition sentences see #10.)

For examples of definitions of descriptive symbols, where 'a', 'b', 'c', 'P', and 'R' are taken as primitive, see preceding examples # 2,3,4,5. 'Q_2' is indefinite, the other defined symbols are definite.

6. Proof and Derivation.

(Hilbert Kap. I,#11, Kap. III, #6; Carnap-Sy.#10.)

The primitive sentences and rules of inference are used for two purposes, namely for proofs and for derivations. A proof shows that a certain sentence is logically true; such a sentence is called demonstrable. A derivation shows that a certain sentence follows logically from other sentences called the premisses; such a sentence is called derivable from the premisses. Neither the premisses nor the sentences derived from them need be logically true; they may refer to empirical facts. By a proof the sentence proved is asserted as true and, moreover, logically true. By a derivation the sentence derived from the premisses is not asserted, but it is merely stated that if the premisses hold the derived sentence must hold too. The definitions of the two procedures explained are as follows:

A proof is a (finite) series of sentences each of which is either a primitive sentence or a definition sentence or directly derivable from one or two sentences which precede it in the series in accordance with a rule of inference or a definition rule. A sentence is called demonstrable if it is the last sentence in a proof.

A derivation with specified premisses is a (finite) series of sentences each of which is either one of the premisses or a primitive sentence or a definition sentence or directly derivable from one or two sentences which precede it in the series in accordance with a rule of inference or a definition rule. A sentence is called derivable from certain sentences if it is the last sentence in a derivation with those sentences as premisses. -- Thus a proof is a special case of a derivation whose class of premisses is null.

Example of a proof.

(In the explanation, '…/---' means that '---' is substituted for '...' in accordance with R1.)

Explanation of the Single Steps	The Proof (as a series of sentences)	Sent. #
PS1	$p \lor p \supset p$	1
PS4	$(p \supset q) \supset (r \lor p \supset r \lor q)$	2
(2) p/p ∨ p	$(p \lor p \supset q) \supset (r \lor (p \lor p) \supset r \lor q)$	3
(3) q/p	$(p \lor p \supset p) \supset (r \lor (p \lor p) \supset r \lor p)$	4
(4) r/~p	$(p \lor p \supset p) \supset (\sim p \lor (p \lor p) \supset \sim p \lor p)$	5
(1) (5) R2	$\sim p \lor (p \lor p) \supset \sim p \lor p$	6
(6) Def. 2	$(p \supset p \lor p) \supset \sim p \lor p$	7
PS2	$p \supset p \lor q$	8

(8) q/p	$p \supset p \vee p$	9
(9) (7) R2	$\sim p \vee p$	10

Thus '~p ∨ p' is demonstrable.

Example of a derivation.

Explanation	Derivation	Sent. #
2 premisses	(x) (P(x) ⊃ Q(x))	1
	P(a)	2
PS5	(x)F(x) ⊃ F(y)	3
(3) R1d, F(x)/P(x) ⊃ Q(x)	(x) (P(x) ⊃ Q(x)) ⊃ (P(y) ⊃ Q(y))	4
(1)(4) R2	P(y) ⊃ Q(y)	5
(5) y/a	P(a) ⊃ Q(a)	6
(2)(6) R2	Q(a)	7

Thus from '(x) (P(x) ⊃ Q(x))' and 'P(a)', 'Q(a)' is derivable.

7. Theorems about Demonstrability and Derivability.

Th. 1. If '(…) ⊃ (---)' is demonstrable, '---' is derivable from '...'.

Th. 2. If '(...) ≡ (---)' is demonstrable, '...' and '---' are derivable from one another.

The following table contains a series of theorems. Part (a) of each states that a certain sentence is demonstrable. Part (b) states the corresponding relationship of derivability according to Th. 1 or 2. (The sentences given in (b) are merely examples: instead of the constants occurring in them any other constants of the same kind may be taken; instead of the sentential constants any other closed sentences; instead of 'P(x)' any other sentence containing the free variable 'x'.)

1. Sentential Calculus. (PM *2-5; Hilbert Kap. I#2)

Theorem	(a) Demonstrable Sentences	(b) The 2nd Sentence is Derivable from the 1st	
3	p ∨ ~p		
4	p ⊃ p	A	A
5	p . q ⊃ p	A . B	A
6	p . q ⊃ q	A . B	B
7	p ⊃ p ∨ q	A	A ∨ B
8	q ⊃ p ∨ q	B	A ∨ B
9	q ⊃ (p ⊃ q)	B	A ⊃ B
10	~p ⊃ (p ⊃ q)	~A	A ⊃ B
11	(p ≡ q) ⊃ (p ⊃ q)	A ≡ B	A ⊃ B
12	(p ≡ q) ⊃ (q ⊃ p)	A ≡ B	B ⊃ A
13	p . ~p ⊃ q	A . ~A	any sentence

Theorem	(a) Demonstrable Sentences	(b) The Last Sentence Is Derivable from the 1st Two		
20	p . (p ⊃ q) ⊃ q	A	A ⊃ B	B
21	p . (p ⊃ q) ⊃ q	A	A ≡ B	B
22	(p ⊃ q) . (q ⊃ r) ⊃ (p ⊃ r)	A ⊃ B	B ⊃ C	A ⊃ C

23	$(p \equiv q) . (q \equiv r) \supset (p \equiv r)$	$A \equiv B$	$B \equiv C$	$A \equiv C$
24	$p . \sim p \supset q$ (like Th. 13)	A	$\sim A$	any sentence

Theorem	(a) Demonstrable Sentences	(b) The Two Sentences Are Derivable from each other	
30	$p \equiv p$	A	A
31	$\sim\sim p \equiv p$	$\sim\sim A$	A
32	$p \vee q \equiv q \vee p$	$A \vee B$	$B \vee A$
33	$p . q \equiv q . p$	$A . B$	$B . A$
34	$p \vee (q \vee r) \equiv (p \vee q) \vee r$	$A \vee (B \vee C)$	$(A \vee B) \vee C$
35	$p . (q . r) \equiv (p . q) . r$	$A . (B . C)$	$(A . B) . C$
36	$p \vee (q . r) \equiv (p \vee q) . (p \vee r)$	$A \vee (B . C)$	$(A \vee B) . (A \vee C)$
37	$p . (q \vee r) \equiv (p . q) \vee (p . r)$	$A . (B \vee C)$	$(A . B) \vee (A . C)$
38	$\sim(p \vee q) \equiv \sim p . \sim q$	$\sim(A \vee B)$	$\sim A . \sim B$
39	$\sim(p . q) \equiv \sim p \vee \sim q$	$\sim(A . B)$	$\sim A \vee \sim B$
40	$\sim(p \supset q) \equiv p . \sim q$	$\sim(A \supset B)$	$A . \sim B$
41	$\sim(p \equiv q) \equiv (p . \sim q) \vee (\sim p . q)$	$\sim(A \equiv B)$	$(A . \sim B) \vee (\sim A . B)$
42	$(p \supset q) \equiv (\sim q \supset \sim p)$	$A \supset B$	$\sim B \supset \sim A$
43	$[p \supset (q \supset r)] \equiv (p . q \supset r)$	$A \supset (B \supset C)$	$A . B \supset C$
44	$(p \equiv q) \equiv (q \equiv p)$	$A \equiv B$	$B \equiv A$
45	$(p \equiv q) \equiv (\sim p \equiv \sim q)$	$A \equiv B$	$\sim A \equiv \sim B$

2.<u>(Lower) Functional Calculus</u>. (PM *9-11; Hilbert Kap III, #6)

Theorem	(a) Demonstrable Sentences	(b) The 2nd Sentence is Derivable from the 1st	
50	$(x)F(x) \supset F(Y)$	$(x)P(x)$	$P(a)$
51	-------------	$P(x)$	$P(a)$
52	$F(y) \supset (\exists x)F(x)$	$P(a)$	$(\exists x)P(x)$
53	$(\exists x)(y)H(x,y) \supset (y)(\exists x)H(x,y)$	$(\exists x)(y)R(x,y)$	$(y)(\exists x)R(x,y)$

		The Last Sentence is Derivable from the First Two	
60	$(x)(F(x) \supset G(x)) . (x)F(x) \supset (x)G(x)$	$(x)(P(x) \supset Q(x))$ $(x)P(x)$	$(x)Q(x)$
61	$(x)(F(x) \supset G(x)) . (\exists x)F(x) \supset (\exists x) G(x)$	$(x)(P(x) \supset Q(x))$ $(\exists x)P(x)$	$(\exists x)Q(x)$
62	$(x)(F_1(x) \supset F_2(x)) . (x) (F_2(x) \supset F_3(x)) \supset (x) ((F_1(x) \supset F_3(x))$	$(x)(P_1(x) \supset P_2(x))$ $(x)(P_2(x) \supset P_3(x))$	$(x) (P_1(x) \supset P_3(x)$

Theorem	(a) Demonstrable Sentences	(b) The Two Sentences Are Derivable from Each Other	
70	-----------------	(x)P(x)	P(x)
71	~(x)F(x) ≡ (∃x)~F(x)	~(x)P(x)	(∃x)~P(x)
72	~(∃x)F(x) ≡ (x)~F(x)	~(∃x)P(x)	(x)~P(x)
73	(x)F(x) ≡ ~(∃x)~F(x)	(x)P(x)	~(∃x)~P(x)
74	(∃x)F(x) ≡ ~(x)~F(x)	(∃x)P(x)	~(x)~P(x)
75	(x)(p ∨ F(x)) ⊃ p ∨ (x)F(x)	(x)(A ∨ P(x)}	A ∨ (x)P(x)
76	(x)(y)H(x,y) ≡ (y)(x)H(x,y)	(x)(y)R(x,y)	{y)(x)R(x,y)
77	(∃x)(∃y)H(x,y) ≡ (∃y)(∃x)H(x,y)	(∃x)(∃y)R(x,y)	(∃y)(∃x)R(x,y)

3. Identity

Identity is reflexive, symmetrical, and transitive

Theorem	(a) Demonstrable Sentences	(b) The last Sentence Is Derivable from the 1st or the 1st Two	
80	x = x	----------	---------
81	x = y ⊃ y = x	a = b	b = a
82	x = y . y = z ⊃ x = z	a = b b = c	a = c

8. Theorems about Replacement.

(Hilbert Kap.III,#7; Carnap-Sy.#13D)

Th. 90. Theorem of Equivalence. The two members of an equivalence may be replaced by each other in any context.

Explanation. Let ‘...’ and ‘---’ be any two sentences, closed or open. Let ‘..(...)..’ be a sentence containing ‘...’ as part and ‘..(---)..’ the corresponding sentence with ‘---’ instead of ‘...’. Then ‘..(---)..’ is derivable from ‘..(...)..’ and ‘(...) ≡ (---)’; and if the equivalence sentence is demonstrable, ‘..(...).:’ and ‘..(---)..’ are derivable from each other. (The derivability stated holds even in the case where ‘...’ contains free variables which are bound in ‘..(...)..’, -- see the third example below.)

(The Theorem of Equivalence holds likewise in any enlarged system, e.g., in the higher functional calculus to be explained below, provided no intensional connective or predicate occurs in the system.)

Examples.

Two Premisses		Derivable from the Two Premisses
$A \equiv B$	$\sim A$	$\sim B$
$A \equiv B$	$C \,.\, (A \vee D)$	$C \,.\, (B \vee D)$
$P_1(x) = P_2(x)$	$A \supset (\exists x)(P_1(x) \vee Q(x))$	$A \supset (\exists x)(P_2(x) \vee Q(x))$

Th. 91. Theorem of Identity. The two members of an identity sentence my be replaced by each other in any context.

Explanation. Let ‘..a..’ be a sentence containing ‘a’, and ‘..b..’ the corresponding sentence containing ‘b’ in place of ‘a’. Then ‘..b..’ is derivable from ‘..a..’ and ‘a = b’. The same holds in any enlarged system where ‘=’ is used not only between individual symbols but also between the expressions of other types, even if they contain free variables, provided sentences analogous to PS 7 and 8 are demonstrable for those other types.

Example. '(∃y)R(b,y)' is derivable from '(∃y)R(a,y)' and 'a = b'.

Theorems 90 and 91 make it possible to formulate definition sentences by means of '≡' and '=' (see #5).

9. Higher Functional Calculus.

(Hilbert Kap.IV #1; Carnap-Abr.#9,13; Carnap-Sy.#27.)

The H.F.C. is characterized by the admission of predicates of higher levels, i.e., those whose arguments may themselves be predicates. Every predicate -- constant or variable -- is assigned to a certain level and within this level to a certain type; for a predicate variable, only predicates of the same type may be substituted. A predicate variable of any type may occur also in a universal or existential operator. The formative and transformative rules stated above for the lower functional calculus (#3,4) would then need to be supplemented accordingly.

Classification of Levels and Types.

The level and the type of a predicate is determined by the level and type of its arguments in the following way:

1. Every individual symbol belongs to the zero-level and the type 0.
2. A series of n arguments which have the types t_1, t_2, ... and t_n respectively, has the type $t_1, t_2, \ldots, t_n$. Its level number is the highest of the level numbers of the arguments.
3. A symbol which is the predicate of a series of arguments of the type t_l and the level m, belongs to the type (t_1) and to the level m + 1.

Example.

Expression (See #1)	The Argument Expression Is of — Type	Level	Therefore the Predicate Is of — Type	Level
P(a)	0	0	(0)	1
R(a,b)	0,0	0	(0,0)	1
T(a,b,c)	0,0,0	0	(0,0,0)	1
^{2}P(P)	(0)	1	((0))	2
^{2}R(P,P)	(0),(0)	1	((0),(0))	2
^{2}S(a,P)	0,(0)	1	(0,(0))	2
^{3}T(^{2}R,a,P)	((0),(0)),0,(0)	2	(((0),(0)),0,(0))	3

10. Functors. (Carnap-Sy. #3, 27)

A functor, like a predicate, has arguments; but unlike a predicate, its full expression (consisting of the n-place functor and n arguments) is not a sentence, but an individual expression, a predicate expression, or a functor expression.

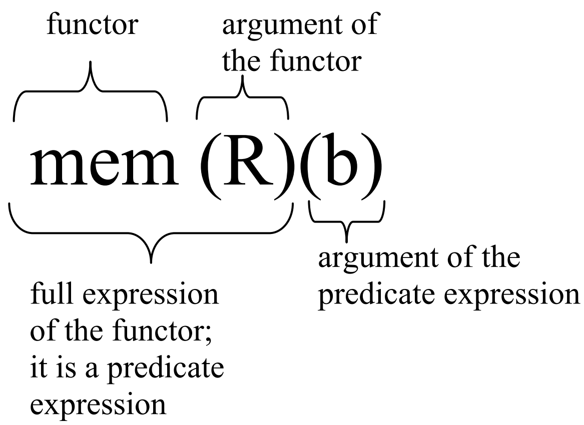
If the series of arguments of a certain functor belongs to the type t_1 and the level m_l, and the full expression belongs to the type t_2 and the level m_2, then we assign the functor itself to the type ($t_1 : t_2$) and its level number is m + 1, where m is the higher of the numbers m_l and m_2.

Examples of Functors.

Full Expression of the Functor	Explanation
mem_1(R)	the first domain of R, i.e., the class (or property) of the 1st-place members of R, i.e., of those objects which bear the relation R to something.
mem_2(R)	the second (or converse) domain of R, i.e., the class of the 2nd-place members of R, i.e., of those objects to which something bears the relation R.
mem(R)	the field of R, i.e., the class of the members of R.

init(R)	the class of the initial members of R, i.e., of those which are first-place but not 2nd-place members of R.
sm(^{2}P)	the class-sum of ^{2}P, i.e., the class of those objects which belong to at least one element-class of ^{2}P.
pr(^{2}P)	the class-product of ^{2}P, i.e., the class of those objects which belong to every element-class of ^{2}P.

Example: 'mem(R)(b)' means: "b has the property mem(R), i.e., is a member of the relation R."



Definitions of the functors explained above:

Def. 10. $\text{mem}_1(H)(x) \equiv (\exists y)H(x,y)$
Def. 11. $\text{mem}_2(H)(x) \equiv (\exists y)H(y,x)$
Def. 12. $\text{mem}(H)(x) \equiv \text{mem}_1(H)(x) \vee \text{mem}_2(H)(x)$
Def. 13. $\text{init}(H)(x) \equiv \text{mem}_1(H)(x) \,.\, {\sim}\text{mem}_2(H)(x)$
Def. 14. $\text{sm}(^2F)(x) \equiv (\exists G)[^2F(G) \,.\, G(x)[$
Def. 15. $\text{pr}(^2F)(x) \equiv (G)[^2F(G) \supset G(x)]$

Determination of type and level for 'mem'. (It is the same for the functors 'mem_1', 'mem_2', and 'init'.)

Expression	Kind of Expression	Type	Level
mem(R)(b)	sentence	---	---
b	individual	0	0
mem(R)	predicate expression	(0)	1
R	predicate	(0,0)	1
mem	functor	((0,0) : (0))	2

NOTES FOR SYMBOLIC IDGIC. SECOND PART

by

Rudolf Carnap

11. PREDICATE EXPRESSIONS; IDENTITY

We use the sentential connectives not only for sentences but also for connecting predicates to compound predicate expressions. If such an expression is followed by arguments it is included in brackets.

Def. 20.	(~F) (x)	≡	~(F (x))
Def. 21.	(F ∨ G) (x)	≡	F (x) ∨ G (x)
Def. 22.	(F . G) (x)	≡	F (x) . G (x)
Def. 23.	(F ⊃ G) (x)	≡	(F (x) ⊃ G (x))
Def. 24.	(F ≡ G) (x)	≡	(F (x) ≡ G (x))

Conventions for Avoiding Brackets (analogous to those on p. 4):

'~' binds more strongly than '|' (#13, Def. 55); this binds more strongly than '∨' and '.'; and these bind more strongly than '⊃' and '≡'. Instead of '(R | S) | T', which is equivalent to 'R | (S | T)' (see #14, Theorem 131), we may write ,R | S | T'.

A predicate expression (or a predicate) in curved brackets is used as an abbreviation for the (closed) universal sentence of that predicate expression. If such a predicate expression included in curved brackets is a separate sentence (i.e., not a part of another sentence) the curved brackets may be omitted. (Whenever such a sentence is inserted into another sentence, e.g., by substitution, the curved brackets must of course be restored.) As an abbreviation for '(∃x)[(...) (x)]', where any predicate expression stands at the place of '...', we write '(∃{…}'. Instead of '({…}) we may write '{…}'.

	Examples	abbreviation for
1. 2.	$\{P\}$ P	$(x)P(x)$
3. 4.	$\{\sim P\}$ $\sim P$	$(x)\sim P(x)$
5.	$\sim\{P\}$	$\sim(x)P(x)$
6. 7.	$\{P \supset Q\}$ $P \supset Q$	$(x)(P(x) \supset Q(x))$
8.	$P_1 . P_2 \equiv Q_1 \vee Q_2$	$(x)[(P_1 . P_2 \equiv Q_1 \vee Q_2)(x)]$ $(x)[P_1 (x) . P_2 (x) \equiv Q_1 (x) \vee Q_2 (x)]$
9.	$\{P_1 . P_2\} \equiv \{Q_1 \vee Q_2\}$	$(x)[(P_1 . P_2 (x)] \equiv (x) [(Q_1 \vee Q_2) (x)]$ $(x)(P_1 (x) . P_2 (x) \equiv (x)(Q_1 (x) \vee Q_2(x))$
10.	$R \supset S$	$(x) (y) (R (x, y) \supset S (x, y))$
11.	${}^2P \supset {}^2Q$	$(F) ({}^2P (F) \supset {}^2Q (F))$
12.	$\exists \{P\}$	$(\exists x) P (x)$
13.	$\sim\exists \{P\} \equiv \{\sim P\}$	$\sim(\exists x) P (x) \equiv (x)\sim P(x)$
14.	$\exists \{R . S\}$	$(\exists x)(\exists y) (R (x,y) . S (x,y))$
15.	$\exists \{{}^2P\}$	$(\exists F) {}^2P (F)$

Explanation in class-terminology. 1. Predicate-expressions: The class P is the complement of the class P; $P \vee Q$ is the sum, $P . Q$ the product of the classes P and Q. 2. Sentences: '$\{P \supset Q\}$' says that the class P is contained in Q (or: P is a sub-class of Q); '$\{P \equiv Q\}$' says that the class P is the same as the class Q.

Instead of introducing identity among individuals by primitive sentences (PS 7 and 8, Part I, p. 5) we may introduce it by a definition (this is possible only in the higher functional calculus):

Def.26. $(x = y) \equiv (F)(F(x) \supset F(y))$ (comp. PM., vol. I. #13)

The same sentences as before are then demonstrable, e.g., PS 7 and 8 (p. 5), and Theorems 80, 81, and 82 (p. 15).

In an analogous way, identity among predicates of any type and identity among functors of any type may be defined:

Def. 27. $({}^{n}F = {}^{n}G) \equiv ({}^{n+1}F)\ [{}^{n+1}F({}^{n}F) \supset {}^{n+1}F({}^{n}G)]$

Def. 28. $({}^{n}f = {}^{n}g) \equiv ({}^{n+1}F)\ [{}^{n+1}F({}^{n}f) \supset {}^{n+1}F({}^{n}g)]$

Here, analogous theorems hold. Also the Theorem of Identity (Th. 91, p. 16) holds for all these kinds of identity.

12. λ -EXPRESSIONS

A λ-expression has the form '(λ x) (...x.…)'; '(λ x)' is called the λ -operator, '...x...' its operand.

1. λ-predicate-expressions. If the operand '...x...' is a sentence the λ-expression is a predicate expression which moans "the property of x such that ...x..." (or: "the class of all x such that ...x.…"). Hence a full sentence '[(λ x) (...x...)] (a)' means the same as '...a...'. Analogously, '(λ x, y)' is used for constructing a two-place predicate expression, '(λ F)' for a second level predicate expression, etc. In accordance with this explanation the system of primitive sentences (#3) would have to be supplemented by

'$[(\lambda x_1, x_2,\ldots x_n)(K(x_1, x_2,\ldots x_n))](u_1, u_2,\ldots u_n) \equiv K(u_1, u_2,\ldots u_n)$',

where n may be any of the numbers 1, 2, etc.('λ x)P(x)' corresponds to '$\hat{x}(\phi x)$' in PM, Vol. I, #20).

In using λ -expressions, careful attention has to be paid to the brackets. We will apply rule 4 (p.4) also in the case of a λ -operator, but not rule 5. That means, the brackets including a whole λ -expression may be omitted; instead of '[(λ x) (P(x))] (a)' one may write '(λ x) (P(x)) (a)'. But the brackets including a λ -operand, e.g. 'P (x)' it the given example, must not be omitted. Consequently, an expression of the form ‘$(\lambda x)(..x..)(a)$’ is to be interpreted as ‘$[(\lambda x)(..x..)](a)$ t and not as ‘$(\lambda x)[(..x..)(a)]$’

	Examples	Formulation in other symbols or in words
1.	$(\lambda x)(P(x))$	P
2.	$(\lambda x)(P(x))(a)$	$P(a)$
3.	$(\lambda x,y)(R(x,y))$	R
4.	$(\lambda x,y)(R(x,y))(a,b)$	$R(a,b)$
5.	$(\lambda x)(P(x) \vee Q(x))$	$P \vee Q$
6.	$(\lambda x)(P(x) \vee (\exists y)R(x,y))(a)$	$P(a) \vee (\exists y)R(a,y)$
7.	$5((\lambda x)(P(x) \vee (\exists y)R(x,y)))$	there are five objects x (see #15) such that $P(x) \vee (\exists y)R(x,y)$.

2. <u>λ-functor-expressions</u>. If the operand ‘...x...’ is not a sentence, the λ-expression is a functor-expression. In this case, as before, the full expression ‘$(\lambda x)(...x...)(a)$’ is to mean the same as ‘...a....’. But in this case, ‘...a...’, and hence also the full expression, is not a sentence; it may be an individual expression, a predicate-expression, or again a functor-expression (see #10, p. 18).

Example. The predicate expression ‘$(\lambda x)(\exists y)R(x,Y)$ t has the same meaning as ‘$\text{mem}_l(R)$’ (see Part I, p. 19). The functor expression ‘$(\lambda H)(\lambda x)(\exists y)H(x,y)$’ has the same meaning as the functor ‘mem_l’.

Distinction between ‘$(\lambda x,y)$’ and ‘$(\lambda x)(\lambda y)$’. ‘$(\lambda x,y)(..x..y..)$’ is a two-place predicate expression; ‘$(\lambda x,y)(..x..y..)(a,b)$’ mean the same as ‘..a..b..’. ‘$(\lambda x)(\lambda y)(..x..y..)$’ is (according to rule 4, not 5) an abbreviation for ‘$(\lambda x)[(\lambda y)(..x..y..)]$’ and hence is a one-place functor expression; ‘$(\lambda x)(\lambda y)(..x..y..)(a)$’ is a one-place predicate expression with the same meaning as ‘$(\lambda y)(..a..y..)$’. Hence

'$(\lambda x)(\lambda y)(..x..y..)(a)(b)$', which is short for '$[[(\lambda x)[(\lambda y)(..x..y..)]](a)](b)$', is a sentence having the same meaning as '..a..b..', and hence the same as the sentence in the first example.

The use of a λ -expression as definiens makes it possible to give any definition of a predicate or a functor the form of an explicit definition in the narrower sense (see p. 7), the definiendum consisting of the defined symbol only. There are many examples of this in the next section.

13. THEORY OF RELATIONS

(Carnap, Abr. #15, 16, 19, 21, 23, 27; PM, vol. I)

In what follows we define some of the frequently used concepts of the theory of relations, and a few of the theory of classes, including some special relations and functors, properties of relations, relations between relations, etc. From a definition schema containing 'n' one gets a series of definitions by putting 'l', '2', '3', etc., one after the other, at the place of 'n'.
In these definitions, 'H', 'H_l', et c. are taken as predicate variables of degree 2; 'K', 'K_1', etc. as predicate variables of any degree n. A predicate of degree n designates an n-place attribute; i.e., if n = 1, a property or class, and if n >1, an n-place relation. For 'mem_l', etc. see Part I, p. 19. As predicates, we take groups of one or more letters beginning with a capital; as functors, those beginning with a small letter.

Definitions.

Number		Definiendum	Definiens	
Def.	30	$\langle x_1; x_2; \ldots x_n \rangle$	$(\lambda u)[u = x_1 \lor u = x_2 \lor \ldots \lor u = x_n]$	
	31	$\langle x_1; x_2; \ldots x_n \rangle$	$(\lambda u_1, u_2, \ldots u_n)[u_1 = x_1 . u_2 = x_2 . \ldots u_n = x_n]$	
	32	V_n	$(\lambda x_1, x_2, \ldots x_n)(x_1 = x_1)$	(Comp. PM, Vol. I, #24,25)
	33	Λ_n	$(\lambda x_1, x_2, \ldots x_n) \sim (x_1 = x_1)$	

Number	Definiendum	Definiens
34	sm_n	$(\lambda N)(\lambda x_1, x_2, ..x_n)(\exists K)[N(K) . K(x_1, x_2, ...x_n)]$
		(PM, #40, 41)
35	pr_n n	$(\lambda N)(\lambda x_1, x_2, ..x_n)(\ K)[N(K) \supset K(x_1, x_2, ...x_n)]$
36	sub_n	$(\lambda K_1)(\lambda K_2)\{K_2 \supset K_1\}$ (PM, #60,61)
40	I_n	$(\lambda x_1, x_2, ..x_n)(x_1 = x_2 . x_2 = x_3 . \ldots . x_{n-1} = x_n]$ (PM, #50)
41	I	I_2
42	J_n	$(\lambda x_1, x_2, ..x_n)[\sim(x_1 = x_2) . \sim(x_1 = x_3) . \ldots . \sim(x_1 = x_n)$ $. \sim(x_2 = x_3) . \sim(x_2 = x_4) . \ldots . \sim(x_2 = x_n)$ $\ldots . \sim(x_{n-2} = x_{n-1}) . \sim(x_{n-2} = x_n)]$ (for $n > 1$)
43	J	J_2
50	H(-,y)	(λx)(H(x,y)) (PM, #32, ‘ $\vec{R}$ ‘y’)
51	H(x,-)	(λx)(H(x,y)) (PM, #32, ‘ $\vec{R}$ ‘x’)
55	$H_1 \mid H_2$	$(\lambda x,y)(\exists u)[H_1(x,u) . H_2(u,y)]$ (PM, #34)
56	K in F	$(\lambda x_1, x_2, ..x_n)[K(x_1, x_2, ...x_n) . F(x_1) . F(x_2) \ldots F(x_n)]$ (PM, #36)
57	H “ F	$(\lambda x)(\exists y)[F(y) . H(x,y)]$ (PM, #37)
60	H^0	I in mem(H)
61	H^1	H
62	H^{n+1}	$H^n \mid H$
63	H^{-n}	$(\lambda x,y)(H^n(y,x)$ (‘H^{-1}’ : PM, #31, ‘ $\breve{R}$ ’)
64	Her	$(\lambda F,H)\ (x)(y)[F(x) . H(x,y) \supset F(y)]$ (PM, #90)
65	$H \geq 0$	$(\lambda x,y)\ [mem(H)(x) . (F)(Her(F,H) . F(x) \supset F(y))]$ (PM, #90, ‘R_*’)
66	$H > 0$	$H \mid H \geq 0$ (PM, #91, ‘R_{po}’)
70	Sym	$(\lambda H)(H \supset H^{-1})$
71	As	$(\lambda H)(H \supset \sim H^{-1})$
72	Trans	$(\lambda H)(H^2 \supset H)$ (PM, II, #201)
73	Intr	$(\lambda H)(H^2 \supset \sim H)$
74	Refl	$(\lambda H)(H^0 \supset H)$
75	Irr	$(\lambda H)(H \supset J)$ (PM, II, #200)

Number	Definiendum	Definiens
76	Reflex	$(\lambda H)(I \supset H)$
77	Connex	$(\lambda H)(J \text{ in mem}(H) \supset H \vee H^{-1})$ (PM II, #202)
78	Ser	Irr. Trans. Connex (PM II, #204)
79	Dense	$(\lambda H)(H \supset H^2)$ (PM II, #270, 'comp')
80	Un_1	$(\lambda H)(H \mid H^{-1} \supset I)$ (PM, #71,72,'1→Cls', 'Cls→1', '1→1')
81	Un_2	$(\lambda H)(H^{-1} \mid H \supset I)$ (PM, #71,72,'1→Cls', 'Cls→1', '1→1')
82	UnUn	$Un_1 . Un_2$ (PM, #71,72,'1→Cls', 'Cls→1', '1→1')
90	$Corr_n$	$(\lambda H, K_1, K_2)$ [UnUn(H) . {mem(K_1) $\supset$ mem_1(H)} . {mem(K_2) $\supset$ mem_2(H)} . $(x_1)(y_1)(x_2)(y_2)\ldots(x_n)(y_n)$ $[H(x_1,y_1) . H(x_2,y_2)\ldots . H(x_n,y_n) \supset (K_1(x_1,x_2,\ldots x_n) \equiv K_2(y_1,y_2\ldots,(y_n))]]$
91	Is_n	$(\lambda K_1, K_2)(\exists H) Corr_n(H, K_1, K_2)$ (PM, I, #73, 'sm'; II, #151, 'smor')
92	Str_n	$(\lambda K_1)(\lambda K_2)(Is_n (K_2, K_1))$ (PM, II, #100, 'Nc'; #152, 'Nr')
93	Str_n	$(\lambda N)(\exists K) \{N \equiv Str_n(K)\}$ (PM, II, #100, 'NC'; #152, 'NR')
94	$Struct_n$	$(\lambda N)(Her(N, Is_n))$

Instead of '$(\langle\ldots\rangle)$' we may write '$\langle\ldots\rangle$'

<u>Explanations</u>. Def. 30. $\langle a_1; a_2; \ldots a_n\rangle$ is the class whose only members are $a_1, a_2 \ldots a_n$. $\langle a\rangle$ is the <u>unit class</u> of a. 31. $\langle a_1, a_2, \ldots a_n\rangle$ is the n-place relation whose only n-ad is $a_1, a_2 \ldots a_n$. 32. V_n is the universal. attribute (i.e. class or relation) of degree n, hence V_1 the <u>universal class</u>. 33.Λ_n is the null attribute, Λ_1 the <u>null class</u>. 34, 35. If M is a class of classes, sm_1(M) is the <u>sum</u> (disjunction) of the classes which are members of M, and pr_1(M) their <u>product</u> (conjunction). If M is a class of two-place relations sm_2(M) is the sum (disjunction) of the relations belonging to M, and pr_2(M) their product (conjunction). 36. sub_1(P) is the class of the subclasses of P; sub_2(R) is the class of the sub-relations of R. 40-43. I is

identity, J diversity, (non-identity) for two arguments, I_3 and J_3 for three arguments, etc. 50, 51. R(-,b) is the class of those objects which have the relation R to b, called the referents of b; R(a,-) is the class of those objects to which a has the relation R, called the rolata of a. 55. R | S is called the relative product of R and S; it holds between a and c if there is an intermediate member b such that R(a,b) and S(b,c). 56. T in P is the sub-relation of T which we get by restricting the field to the class P. 57. R " P is the class of those objects which have the relation R to some member of P. 60-62. The powers of a relation: R^0 is identity among members of R; R^1 is R itself; R^2 is R | R; R^3 is R^2 | R, etc. 63. R^{-1} is called the converse of R; it holds for the same pairs of members as R, but in the inverse order. R^{-2} is the converse of R^2, etc. 64. 'Her(P,R)' means that the property P is hereditary with respect to R, i.e., that it is transferred from any member a to any other member b to which a has the relation R. 65,66. Ancestral relations of first and second kind. '$R^{\geq 0}$(a,b)' means that a is a member of R and b has all hereditary properties of a. This is the case if there is a finite number $n \geq 0$ such that R^n(a,b). '$R^{>0}$(a,b)I holds if there is a finite number $n > 0$ such that R^n(a,b). Definitions 70 to 82 define second-level predicates which designate properties of two-place relations. 70. 'Sym(R)' means that R is symmetrical, i.e., if R holds for a pair of members then it holds for the inverse pair also. (Examples: cousin, similar, parallel). 71. 'As(R)': R is asymmetrical, i.e., R does not hold in any pair in both directions. (Examples: father, <). 72. 'Trans(R)': R is transitive, i.e., if R(x,y) and R(y,z) then R(x,z). (Ex.: ancestor, parallel, <). 73. 'Intr(R)': R is intransitive, i.e. if R(x,y) and R(y,z) then not R(x,z). (Ex.: father). 74. 'Refl(R)': R is reflexive, i.e. if x is a member of R then R(x,x). (Ex.: equal in weight, similar, ≤). 75. 'Irr(R)': R is irreflexive, i.e., if x is a member of R then not R(x,x). (Ex.: father, brother, <). 76. 'Reflex(R) '; R is totally reflexive, i.e., for every x, R(x,x).

(Ex.: equal in weight, similar). 77. ‘Connex(R)’: R is connected, i.e., R holds in any pair of two different members of R in at least one of the two directions. (Ex.: $<$, $\leq$). 78. ‘Ser(R)’: R is a series, i.e., R is irreflexive, transitive, and connected. (Ex.: $<$). 79. ‘Dense(R)’: R is dense, i.e., for any x and y such that R(x,y) there is an intermediate member u such that R(x,u) and R(u,y). (Ex.: $<$among fractions). 80. ‘Un_l(R)’: R is a one-many relation, or: R is univocal with respect to the first argument, i.e., no two different members have the relation R to any member. (Ex.; father, square). 81. ‘Un_2(P)’: R is a many-one relation, or: R is univocal with respect to the second argument, i.e., no member has the relation R to more than one member. (Ex.: square root). 82. ‘UnUn(R)’:R is a one-one relation, or: R is univocal in both directions, i.e., R is both one-many and many-one. (Ex.: successor among natural numbers.)

90. ‘$Corr_n$(R,T_l,T_2)’: R is a correlator for the n-place relations T_1 and T_2, i.e., R is one-one, the members of T_1 are first-place members of R, the members of T_2 are second-place members of R, an n-ad of T_l is correlated by R with an n-ad of T_2 and vice versa. 91. ‘Is_n(T_1,T_2)’: T_1 is isomorphic with T_2, i.e., there is a correlator for T_1 and T_2. ‘Is_l(P_1,P_2)’: the classes P_1 and P_2 are isomorphic (i.e. equal in number). 92. str_n(T) is the (n-place) structure of the (n-place) relation T, i.e., the class of those relations which are isomorphic with T. str_1(P) is the cardinal number of the class P. 93. ‘Str_n(M)’ means that the (second-level) class M is an n-place structure, i.e., the n-place structure of some n-place relation. ‘Str_l(M)’: M is a oneplace structure, i.e., a cardinal number. 94. ‘$Struct_n$(M)’: the (second-level) property M is a structural property of n-place relations, i.e., if T is M, then any relation isomorphic with T is also M. ‘$Struct_1$ M)’: M is a structural property of classes, i.e., a property of classes dependent only upon their cardinal number.

14. THEOREMS IN THE THEORY OF RELATIONS

On each line, the sentences in columns (a) and (b) are derivable from one another; if no sentence is written in (b), the sentence in (a) is demonstrable.

Theorem No.	(a)	(b)
100	$\langle a \rangle (b)$	$b = a$
101	$\langle a_1 ; a_2 \rangle$ (b)	$b = a_1 \vee b = a_2$
102	$\langle a_1 , a_2 \rangle$ (b_1,b_2)	$b_1 = a_1 . b_2 = a_2$
103	$v_1(x)$	
104	$P \equiv V_1$	{P}
105	$F \supset V_1$	
106	V_2 (x,y)	
107	$\sim \Lambda_1(x)$	
108	$P \equiv \Lambda_1$	$\sim \exists$ {P}
109	$\Lambda_1 \supset F$	
110	$\sim \Lambda_2(x,y)$	
111	$\Lambda_n \equiv \sim V_n$	
112	$sm_1(^2P)$ (a)	$(\exists F) (^2P(F) . F (a))$
113	$pr_1(^2P)$ (a)	$(F) (^2P) (F) \supset F(a))$
114	$sub_1(P)$ (Q)	$Q \supset P$
115	$sub_2(R)$ (S)	$S \supset R$
120	I(a,b)	a = b
121	$1_3(a,b,c)$	a = b . b = c
122	J(a,b)	$\sim(a = b)$
123	$J_3(a,b,c)$	$\sim(a = b) . \sim(a = c) . \sim(b = c)$
130	(R | S) (a,b)	$(\exists y)$ (R(a,y) . S(y,b))
131	$(H_1 \mid H_2) \mid H_3 \equiv H_1 \mid (H_2 \mid H_3)$	

Theorem No.	(a)	(b)
132	(R in P) (a,b)	R(a,b) . P(a) . P(b)
133	(R “ P) (a)	$(\exists y)(P(y) . R(a,y))$
140	$R^0(a,b)$	a = b . mem(R) (a)
141	$R^1(a,b)$	R(a,b)
142	$R^2(a,b)$	$(\exists y)(R(a,y) . R(y,b))$
143	$R^{-1}(a,b)$	R(b,a)
144	$(R^{-1})^{-1}(a,b)$	R(a,b)
150	Her(P,R)	$(x)(y) [P(x) . R(x,y) \supset P(y)]$
151	$R^{\geq 0}(a,b)$	$mem(R)(a) . (F) [Her(F,R) . F(x) \supset F(y)]$
152	$H^0 \supset H^{\geq 0}$	
153	$H \supset H^{\geq 0}$	
154	$H^2 \supset H^{\geq 0}$ etc.	
155	$R^{>0}(a,b)$	$(\exists y) [R(a,y) . R^{\geq 0}(y,b)]$
156	$H \supset H^{>0}$	
157	$H^2 \supset H^{>0}$	
158	$H^3 \supset H^{>0}$	
	etc.	
159	$H^{>0} \supset H^{\geq 0}$	
160	$R^{\geq 0}(a,b)$	$R^{>0}(a,b) \vee R^0(a,b)$
170	Sym(R)	$R \supset R^{-1}$
171	Sym(R)	$(x)(y)(R(x,y) \supset R(y,x))$
172	As(R)	$R \supset \sim R^{-1}$
173	As(R)	$(x)(y)(R(x,y) \supset \sim R(y,x))$
174	Trans (R)	$R^2 \supset R$
175	Intr(R)	$R^2 \supset \sim R$
176	Refl (R)	$R^0 \supset R$
177	Refl(R)	$mem(R)(x) \supset R(x,x)$

Theorem No.	(a)	(b)
178	$Irr(R)$	$R \supset J$
179	$Irr(R)$	$(x){\sim}R(x,x)$
180	$Trans.Sym \supset Refl$	
181	$As \supset Irr$	
182	$Trans \supset (As \equiv Irr)$	
183	$Reflex(R)$	$I \supset R$
184	$Reflex(R)$	$(x)R(x,x)$
185	$Reflex(R)$	$Refl(R) . \{mem(R)\}$
186	$Connex(R)$	$mem(R)(x) . mem(R)(y) . {\sim}(x = y) \supset R(x,y) \vee R(y,x)$
187	$Ser(R)$	$(Irr . Trans . Connex)(R)$
188	$Ser(R)$	$(As . Trans . Connex)(R)$
189	$Dense(R)$	$R \supset R^2$
190	$Dense (R)$	$R(x,z) \supset (\exists y)(R(x,y) . R(y,z))$
191	$Un_1(R)$	$R \mid R^{-1} \supset I$
192	$Un_1(R)$	$R(x,z) . R(y,z) \supset x = y$
193	$Un_2(R)$	$R^{-1} \mid R \supset I$
194	$Un_2(p)$	$R(x,y) . R(x,z) \supset y = z$
195	$Un_2(R)$	$Un_1(R^{-1})$
196	$UnUn(R)$	$Un_1(R) . Un_2(R)$
197	$UnUn(R)$	$UnUn(R^{-1})$
210	$Corr_1(R,P_1,P_2)$	$UnUn(R) . \{P_1 \supset R \text{ “ } P_2\} . \{P_2 \supset R^{-1} \text{ “ } P_1 \}$
211	$Corr_2(R,S_1,S_2)$	$UnUn(R) . \{mem(S_2) \supset mem_2(R)\} . \{S_1 \equiv R \mid S_2 \mid R^{-1}\}$
212	$Is_1(P_1,P_2)$	$(\exists H)\ Corr_1(H,P_1,P_2)$
213	$Is_2(S_1,S_2)$	$(\exists H)\ Corr_2(H,S_1,S_2)$
214	$str_n(T_1)(T_2)$	$Is_n(T_2,T_1)$
21$_5$	$Str_n\ (^2P)$	$(\exists K)\{^2P \equiv str_n(K)\}$
216	$Struct_n(^2P)$	$Her(^2P,Is_n)$
217	$Struct_n(^2p)$	$(K_1)(K_2)[^2P(K_2) . Is_n(K_1,K_2) \supset {}^2P(K_2)]$

15. CARDINAL NUMBERS

(PM, vol. II, Part III; Carnap-Abr. #19, 21,26)

The cardinal numbers are the (one-place) structures of classes. We will here define some finite cardinal numbers (0,1,2,etc.), the class of progressions (Prog), and the smallest transfinite cardinal number, λ^{λ}_0 (aleph-zero); this is the cardinal number of any progression. If λ^{λ}_0 (P), P is called a denumerable class.

'3_m(P)' is to mean: "there are at least 3 P-objects"; '3(P)': "P has the cardinal number 3"; i.e., "there are 3 P-objects". For 'mem' and 'init' see Part I, p. 19.

Definitions.

Number	definiendum	definiens
100	1_m	$(\lambda F)\ (\exists \{F\})$
101	2_m	$(\lambda F)\ (\exists \{J_2\ \mathrm{in} F\})$
102	3_m	$(\lambda F)\ (\exists \{J_3\ \mathrm{in} F\})$
	etc.	
105	0	$\sim 1_m$
106	1	$1_m\ .\ \sim 2_m$ (Comp. PM, vol. II, #101)
107	2	$2_m\ .\ \sim 3_m$
108	3	$3_m\ .\ \sim 4_m$
	etc.	
110	Prog	$(\lambda H)\ [\mathrm{UnUn}(H)\ .\ 1(\mathrm{init}(H))\ .\ 0(\mathrm{init}(H^{-1}))\ .\ \mathrm{Connex}(H^{>0})]$ (PM #122)
111	λ^{λ}_0	$(\lambda F)\ (\exists H)[\mathrm{Prog}(H)\ .\ \{F \equiv \mathrm{mem}(H)\}]$ (PM #123)
120	$N_1 + N_2$	$(\lambda F)\ (\exists G_1)(\exists G_2)\ [\{F \equiv G_1 \vee G_2\}\ .\ \sim\exists \{G_1\ .\ G_2\}\ .\ N_1(G_1)\ .\ N_2(G_2)]$

Number	definiendum	definiens
121	Pred	$(\lambda N_1,N_2)\{N_1+1 \equiv N_2\}$
122	Fin	$(\lambda N)\ (\mathrm{Pred} \geq^{0}(0,N))$

Explanations to Def. 120-122. A class F has the cardinal number $N_1 + N_2$, the (arithmetical) sum of N_1 and N_2, if F can be divided into two mutually exclusive subclasses G_1 and G_2 which have the cardinal numbers N_1 and N_2 respectively. N_1 is the predecessor of N_2 if N_2 is N_1+1. N is a finite cardinal number if 0 has the relation Pred $\geq^{0}$ to N.

Theorems. The sentences (a) and (b) are derivable from one another; where no sentence (b) is given, sentence (a) is demonstrable.

Theorem No.	(a)	(b)
230	$0(P)$	$\sim\exists\{P\}$
231	$0 = \mathrm{str}_1(\Lambda_1)$	
232	$0(P)$	$P \equiv \Lambda_1$
233	$1(P)$	$(\exists x)(y)(P(y) \equiv x = y)$
234	$1(P)$	$(\exists x)\{P \equiv \langle x\rangle\}$
235	$1 \equiv \mathrm{Str}_1\ \langle x\rangle$	$(\exists x)(\exists y)(P(x)\ .\ P(y)\ .\ \sim(x = y))\ .\ \sim(\exists x)(\exists y)(\exists z)$ $(P(x)\ .\ P(y)\ .\ P(z)\ .\ \sim(x = y)\ .\ \sim(x = z)\ .\ \sim(y = z))$
236	$2(P)$	$(\exists x)(\exists y)(J(x,y)\ .\ \{P \equiv \langle x, y\rangle\})$
237	$2(P)$	$\mathrm{str}_1\langle a, b\rangle \equiv 2$
238	$J(a,b)$	
244	$\mathrm{Str}_1(0)$	
245	$\mathrm{Str}_1(1)$	
246	$\mathrm{Str}_1(2)$	
	etc.	
250	$\mathrm{Prog}(H) \supset \lambda^{\lambda}_{0}\ (\mathrm{mem}(H))$	

Principle of Infinity. Several formulations: 1. “There is an infinite number, i.e., at least a denumerable class, of individuals”: ‘∃ {λ^{λ}_{0}}’. 2. “For any finite cardinal number there is a class (of individuals) having that cardinal number”: “Fin(N) ⊃ ∃ {N}. 3. “There is a progression of individuals”: ‘∃ { Prog}’. (Comp. PM, vol. II, #125). The following sentences are derivable from the Principle of Infinity and hence demonstrable if some formulation of this principle is taken as a primitive sentence : ‘Str_2(Prog)’, ‘Str_1(λ^{λ}_{0})’ i.e., Prog is a structure of relations,) λ^{λ}_{0} is a cardinal number.

16. DESCRIPTIONS

(PM, vol. I, .#14, 30; Carnap-Abr. #7b, l4a; Carnap-Sy. #38c)

An expression of the form ‘(ιx) (...x...) ‘ is to mean, “the object x such that ...x...”. ‘(ιx)’ is called a ι-operator (iota-operator), ‘...x...’ its operand; the expression ‘(ιx) (...x,..)’ is called a description; ‘x’ is bound within the description. Descriptions are used chiefly as arguments for predicates. According to rule 4, p. 4, we may omit the brackets including a description. Example: ‘Q(ιx) (..x..)’. This sentence is to mean: “the object x such that ..x.., is Q,” or more exactly: “there is exactly one x such that ..x.., and this x (or, in other words: every x such that ..x..) is Q.” (Example: “the brother of b is ill”: ‘Ill(ιx)Br(x,b)’). In accordance with this explanation the following may be stated as a primitive sentence for the introduction of the ι-operator: ‘G(ιx) (F(x)) = 1(F) . { F ⊃ G}’.

‘R ′ b’ is used as an abbreviation for the description ‘(ιx)R(x,b)’. Hence it means “the object which has the relation R to b”. (Example: ‘Br ′ b’, “the brother of b”).

It is to be noticed that the rule of substitution does not permit the substitution of a description of either kind (‘(ιx)(P(x))’ or ‘R ′ b’) for a variable.

Theorems. The sentences in each of the following pairs are derivable from one another.

Theorem No.	(a)	(b)
260	Q(ιx) (P(x))	1(P) . {P ⊃ Q}
261	~Q(ιx) (P(x))	0(P) ∨ 2_m(P) ∨ ∃ {P . ~Q}
262	(ιx) (P(x)) = a	1(P) . P(a)
263	(ιx) (P(x)) = a	P ≡ ⟨a⟩
264	(ιx) (P(x)) = (ιx) (Q(x))	1(P) . 1(Q) . {P ≡ Q}
265	Q(R ′ b)	1(R(-,b)) . {R(-,b) ⊃ Q}

ERRATA

First Part.

p. 3, below, and p. 4 above. Cross out the whole content of the column "Kind of Sentences."
p. 4, 1 ne 1. Instead of "(x) (P (x)) $\vee$ (R (x, b))" read "(x)((P (x)) $\vee$ (R (x,b))".
p. 4, after line 4, add: "A sentence is called atomic if it has one of the forms 1, 2, 3; molecular, if no operator occurs; general, if an operator occurs."
p. 4, rule 4. Instead of "operator with operand " read "operator (of any kind) with operand."
p. 4, rule 5. Instead of "operand" read "operand (except with a λ- operator)."
p. 4, below, Example 2a, last column,

instead of:	"A $\vee$ B	read: "~A $\vee$ B
	A . B	~A . B
	A $\supset$ B	~A $\supset$ B
	A $\equiv$ B"	~A $\equiv$ B"

p. 5, line 8 from below, instead of "premiss" read "premisses".
p. 6, below. Cross out "instead of 'x' any other individual variable may be taken" and add the following:
"R4. Rule of Bound Variables. S_3 is constructed out of S_l by replacing an individual variable in an operator and at all places where it occurs as a variable which is free within the operand belonging to the operator, by another individual variable which does not occur in the original operand."
p. 7, after R3b, add the following:

"R4 | (u) P (u) | (v) P (v)"

p. 18, line 2 of #10, instead of "then-n-place read "the n-place."

Second Part.
p. 32, Def.110. Instead of "Prog." read "Prog".
p. 35, line 1. Instead of "sentence" read "sentences".
The spaces left between a predicate and its arguments or between an operator and its operand at many places in the Second Part (esp. on p.20, 21, 34, 35, 36) are of course unnecessary.

TABLE OF CONTENTS

First Part
1. Use of Letters 1
2. Sentential Calculus 2
3. Lower Functional Calculus 3
4. Transformative Rules 5
5. Definitions 7
6. Proof and Derivation 9
7. Theorems about Demonstrability and Derivability 11
8. Theorems about Replacement 16
9. Higher Functional Calculus 17
10. Functors 18

Second Part
11. Predicate Expressions; Identity 20
12. λ-Expressions 22
13. Theory of Relations 24
14. Theorems in the Theory of Relations 29
15. Cardinal Numbers 32
16. Descriptions 34