Journal of Research in Philosophy and History
Vol. 5, No. 1, 2022
www.scholink.org/ojs/index.php/jrph
ISSN 2576-2451 (Print) ISSN 2576-2435 (Online)

Original Paper

Parenthesis Notation

Guoping Du[1,2*]

[1] Institute of Philosophy, Chinese Academy of Social Sciences, Beijing, China

[2] Sichuan Normal University, Chengdu, China

[*] Guoping Du, E-mail: dgpnju@163.com

Received: January 22, 2022 Accepted: February 13, 2022 Online Published: February 22, 2022

doi:10.22158/jrph.v5n1p44 URL: http://dx.doi.org/10.22158/jrph.v5n1p44

Abstract

The formal language of a logical system usually contains several types of symbols. In infix notation, two different kinds of symbols are used to construct compound formula and to indicate the order of combination. The logical constants such as ¬, ∨ are used to construct compound formula, and auxiliary symbols such as ( ) are used to indicate the order of a combination. In Polish notation, there is no need for auxiliary symbols such as ( ), and only one class of symbols, N, C, K, etc., is used as a conjunction to make it function as a parenthesis. Contrary to Polish notation, a new parenthesis notation is put forward in this paper. Parenthesis notation uses only parentheses, and empowers them the function of connectives. More importantly, it is proved in this paper that we can define logical constants such as propositional connectives, quantifiers, modal operators and temporal operators in the same formula by using only parenthesis, which can greatly simplify the initial connectives needed to construct the formal system.

Keywords

parenthesis notation, infix notation, Polish notation, logical constants

Appropriate notation can express the reasoning relationship in a concise and clear way and describe the logic rules more efficiently. This paper proposes a new notation which is different from the existing systems.

1. Introduction

Logical constants such as propositional connectives, quantifiers and modal operators are one of the core issues in logic research. Using appropriate symbols to notate these logical constants can reveal their reasoning characteristics more accurately. Parenthesis notation is a notation system that uses only one pair of parentheses to notate all logical constants.

The notation of logical constants can be traced back at least to William of Shyreswood in England and Peter in Spain in the middle ages. They used A, E, I and O to represent logical constants "universal affirmative", "universal negative", "particular affirmative" and "particular negative" respectively (Bonevac, 2012).

Since the 20th century, there are three common ways to notate logical constants: infix notation, prefix notation and postfix notation. The infix notation is to put the binary propositional connectives in the middle of the two propositions. For example, it uses symbols "$P \wedge Q$", "$P \vee Q$ ", "$P \rightarrow Q$", "$P \leftrightarrow Q$" to notate the relations of "conjunction", "disjunction", "implication" and "equivalence" between propositions P and Q. The infix notation is a very widely used notation.

Prefix notation is to put the connectives before the proposition it connects. This is a unique notation proposed by Lukaswicz and others, so it is also called Polish notation. Prefix notation uses "N*pq*", "C*pq*", "K*pq*", "A*pq*", "E*pq*", "Π*pq*" to express the "Negation" of proposition p and the "implication", "conjunction", "disjunction" and "equivalence" between proposition p and q (Łukasiewicz, 1966).

Postfix notation, also known as reverse Polish notation, places the connectives after the proposition it connects.

Different from the above three notations, parenthesis notation does not use infix notation symbols such as $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$, nor does it use prefix notation symbols such as N, C, K, A, E and Π. It only uses a pair of parentheses to notate various logical constants without using other symbols. For example, the propositional connectives such as "negation $\neg$", "conjunction $\wedge$", "disjunction $\vee$", "implication $\rightarrow$" and modal operators such as "necessity $*$" can be defined simultaneously by using only "( ) ". For example, to define a triple with only parentheses:

$$(PQR) =_{def} [\neg P \vee \neg Q] \wedge * [Q \rightarrow R] \text{ (Note 1).}$$

Then, we obtain the following triple according to the definition:

$$(PQQ) = [\neg P \vee \neg Q] \wedge * [Q \rightarrow Q] = [\neg P \vee \neg Q].$$

$[\neg P \vee \neg Q]$ is a Sheffer function, which can notate all propositional connectives including negation and conjunction, so triple (PQR) can define all propositional connectives.

And we can also obtain the following triple according to the definition:

$$(R \neg RR) = [\neg R \vee \neg\neg R] \wedge * [\neg R \rightarrow R] = * [\neg R \rightarrow R] = * R.$$

Hence, triple (PQR) can define the modal operator "necessary $*$".

It can be seen from this example that in parenthesis notation, all propositional connectives and modal connective "necessary $*$" can be defined at the same time by using only parenthesis and an expression.

2. The Characteristics of Parenthesis Notation

a. Parenthesis notation has strong expression ability. For all propositional connectives, quantifier (universal quantifier $\forall$), modal operator (necessary $*$) and temporal operators (forever G, always H), we can use one expression with only parentheses to define them at the same time. For example, define an octet with only parentheses:

$$(PQRSTUxV)=_{def} [\neg P\vee\neg Q]\wedge*[Q\rightarrow R]\wedge G[R\rightarrow S]\wedge H[S\rightarrow T]\wedge\forall x[U[x]\rightarrow V[x]].$$

According to the above octet, we can obtain the following:

(1) $(QQQQQUxU)$

$= [\neg Q\vee\neg Q]\wedge*[Q\rightarrow Q]\wedge G[Q\rightarrow Q]\wedge H[Q\rightarrow Q]\wedge\forall x[U[x]\rightarrow U[x]]$

$= \neg Q$

Hence, octet $(PQRSTUxV)$ can define propositional connective "negation ¬".

(2) $(\neg P\neg QQQQUxU)$

$= [\neg\neg P\vee\neg\neg Q] \wedge*[Q\rightarrow Q]\wedge G[Q\rightarrow Q]\wedge H[Q\rightarrow Q]\wedge\forall x[U[x]\rightarrow U[x]]$

$= [\neg\neg P\vee\neg\neg Q]$

$= P\vee Q$

Hence, octet$(PQRSTUxV)$ can define propositional connective "disjunction ∨".

As we all know, with negation and disjunction, we can define all the other propositional connectives. According to (1) and (2), all propositional connectives can be defined by octet $(PQRSTUxV)$.

(3) $(Q\neg QQQQUxU)$

$= [\neg Q\vee\neg\neg Q]\wedge*[\neg Q\rightarrow Q]\wedge G[Q\rightarrow Q]\wedge H[Q\rightarrow Q]\wedge\forall x[U[x]\rightarrow U[x]]$

$= *[\neg Q\rightarrow Q]$

$= *Q$

Hence, octet $(PQRSTUxV)$ can define modal operator "necessary ∗".

(4) $(S\neg S\neg SSSUxU)$

$= [\neg S\vee\neg\neg S]\wedge*[\neg S\rightarrow\neg S]\wedge G[\neg S\rightarrow S]\wedge H[S\rightarrow S]\wedge\forall x[U[x]\rightarrow U[x]]$

$= G[\neg S\rightarrow S]$

$= GS$

Hence, octet $(PQRSTUxV)$ can define temporal operator "forever G".

(5) $(S\neg S\neg S\neg SSUxU)$

$= [\neg S\vee\neg\neg S]\wedge*[\neg S\rightarrow\neg S]\wedge G[\neg S\rightarrow\neg S]\wedge H[\neg S\rightarrow S]\wedge\forall x[U[x]\rightarrow U[x]]$

$= H[\neg S\rightarrow S]$

$= HS$

Hence, octet $(PQRSTUxV)$ can define temporal operator "always H".

(6) $(\neg TTTTT\neg VxV)$

$= [\neg\neg T\vee\neg T]\wedge*[T\rightarrow T]\wedge G[T\rightarrow T]\wedge H[T\rightarrow T]\wedge\forall x[\neg V[x]\rightarrow V[x]]$

$= \forall x[V[x]]$

Hence, octet $(PQRSTUxV)$ can define "universal quantifier ∀".

It can be seen from the above definition that all logical constants contained in a formal system can be defined at the same time by only using parentheses with an expression (Note 2).

b. Parenthesis notation is clear.

(1) It has a clear hierarchy. For propositional connectives, we can only use (*QR*) to notate. Even though they are equivalent to Sheffer's in expressive ability, they do not need to add other auxiliary symbols to distinguish the order of combination.

(2) It has a clear scope. In the above-mentioned octet (*PQRSTUxV*), the left parenthesis' variables from left to right are *P*, *Q*, *R*, *S*, *T*, *U*, *x*, *V*, and the right parenthesis' variables from right to left are *V*, *x*, *U*, *T*, *S*, *R*, *Q*, *P*, which are determined by the sequence of symbol strings in the parenthesis.

c. The symbol of parenthesis notation is concise.

In parenthesis notation, the propositional connectives as logical constants can be defined simultaneously by a formula containing only parenthesis, in which only one pair of parentheses is used as the initial connectives, and there is no need for multiple initial symbols or other auxiliary symbols to indicate the order of combination.

3. To Build a Logical System Based on Parenthesis Notation

Next, Modal Propositional Logic is taken as an example to illustrate the construction of formal language and its related system based on parenthesis notation.

DEFINITION 1.

Formal language L_{MP} only contains the following two types of symbols:

(1) Proposition symbols: p_1, p_2, …, p_n, p_{n+1}, …

(2) Connective symbols: (,).

In formal language L_{MP}, the initial connectives are only one pair of parentheses "( )".

DEFINITION 2.

Formulas in formal language L_{MP} are obtained if and only if the following rules are used for a finite number of times:

(1) A single proposition symbol is a formula;

(2) If *P*, *Q*, *R* are formulas, then (*PQR*) is a formula.

It is common to use capital letters, *P*, *Q*, *R*, etc. to indicate any formula. The set of all formulas in L_{MP} is noted as *Form*(L_{MP}).

Some abbreviations are introduced as follows:

DEFINITION 3.

(1) ⮜*P*⮞ $=_{def}$ (*PPP*)

(2) 『*PQ*』 $=_{def}$ (*PQQ*)

(3) <*PQ*> $=_{def}$ (⮜*P*⮞⮜*Q*⮞⮜*Q*⮞) $=_{def}$ ((*PPP*)(*QQQ*)(*QQQ*))

(4) 【*P*】 $=_{def}$ (*P*⮜*P*⮞*P*)$=_{def}$ (*P*(*PPP*)*P*)

(5) 〖*P*〗 $=_{def}$ ⮜【⮜*P*⮞】⮞

DEFINITION 4.

A binary ⟨*W*, *R*⟩ is a frame, if and only if, *W* is any nonempty set and *R* is a binary relation on *W*, that is $R \subseteq W \times W$.

DEFINITION 5.

Assume that $\langle W, R\rangle$ be any frame, V be an assignment to the formula in $Form(L_{MP})$ on $\langle W, R\rangle$, if and only if V is a mapping of the Cartesian product of $Form(L_{MP})$ and W onto the set {1, 0}, that is

$$V: Form(L_{MP}) \times W \rightarrow \{1, 0\}$$

And the following conditions are satisfied: for any formula P, Q, R in $Form(L_{MP})$, for any $w \in W$, $V[(PQR), w] = 1$, if and only if, $V[P, w] = 0$ or $V[Q, w] = 0$, and for any $w' \in W$, if Rww', then $V[Q, w'] = 0$ or $V[R, w'] = 1$.

DEFINITION 6.

Triple $\langle W, R, V\rangle$ is a L_{MP} model, if and only if $\langle W, R\rangle$ is a frame and V is an assignment to the formula in $Form(L_{MP})$.

Theorem 1 Let the triples $\langle W, R, V\rangle$ be any L_{MP} model, for any $w \in W$, $V[\prec P \succ, w] = 1$, if and only if $V[P, w] = 0$.

Proof. According to definition 3 to definition 6, we have

$$V[\prec P \succ, w] = 1 \text{ if and only if } V[(PPP), w] = 1$$

If and only if $V[P, w] = 0$ or $V[P, w] = 0$, and for any $w' \in W$, if Rww', then $V[P, w'] = 0$ or $V[P, w'] = 1$.

If and only if $V[P, w] = 0$ or $V[P, w] = 0$.

If and only if $V[P, w] = 0$.

Theorem 2 Assume Triple $\langle W, R, V\rangle$ is any L_{MP} model, for any $w \in W$, $V[$『PQ』$, w] =1$ if and only if $V[P, w] = 0$ or $V[Q, w] = 0$.

Proof. According to definition 3 to definition 6, we have

$$V[\text{『}PQ\text{』}, w] =1 \text{ if and only if } V[(PQQ), w] =1$$

If and only if $V[P, w] = 0$ or $V[Q, w] = 0$, and for any $w' \in W$, if Rww', then $V[Q, w'] = 0$ or $V[Q, w'] =1$.

If and only if $V[P, w] = 0$ or $V[Q, w] = 0$.

Theorem 3 Assume Triple $\langle W, R, V\rangle$ is any L_{MP} model, for any $w \in W$, $V[<PQ>, w] = 1$ if and only if $V[P, w] = 1$ or $V[Q, w] = 1$.

Proof. According to definition 3 to definition 6, we have

$$V[<PQ>, w] = 1 \text{ if and only if } V[(\prec P \succ \prec Q \succ \prec Q \succ), w] = 1$$

If and only if $V[\prec P \succ, w] = 0$ or $V[\prec Q \succ, w] = 0$, and for any $w' \in W$, if Rww', then $V[\prec Q \succ, w'] = 0$ or $V[\prec Q \succ, w'] = 1$.

If and only if $V[P, w] = 1$ or $V[Q, w] = 1$, and for any $w' \in W$, if Rww', then $V[Q, w'] = 1$ or $V[Q, w'] = 0$.

If and only if $V[P, w] = 1$ or $V[Q, w] = 1$.

Theorem 4 Assume Triple $\langle W, R, V\rangle$ is any L_{MP} model, for any $w \in W$, $V[$【P】$, w] = 1$ if and only if for any $w' \in W$, if Rww', then $V[P, w'] = 1$.

Proof. According to definition 3 to definition 6, we have

V [【P】, w] = 1 if and only if V [(P≺P≻P) , w] = 1,

If and only if V [P, w] = 0 or V [≺P≻, w] = 0, and for any $w' \in W$, if Rww', then V [≺P≻, w'] = 0 or V [≺P≻, w'] = 1.

If and only if V [P, w] = 0 or V [P, w] = 1, and for any $w' \in W$, if Rww', then V [P, w'] = 1 or V [P, w'] = 1.

If and only if for any $w' \in W$, if Rww', then V [P, w'] = 1.

Theorem 5 Assume Triple ⟨W, R, V⟩ is any L_{MP} model, for any $w \in W$, V [〖P〗, w] = 1 if and only if there exists $w' \in W$, Rww', and V [P, w'] = 1.

Proof. According to definition 3 to definition 6, we have

V [〖P〗, w] = 1 if and only if V [≺【≺P≻】≻, w] = 1

If and only if V [【≺P≻】, w] = 0

If and only if there exists $w' \in W$, Rww', and V [≺P≻, w'] = 0.

If and only if there exists $w' \in W$, Rww', and V [P, w'] = 1.

From Theorem 1 to Theorem 5, it can be seen that the abbreviation symbol≺P≻ in definition 3 is equivalent to the negation of classical propositional connective $\neg P$, the symbol『PQ』is equivalent to the alternative denial of Sheffer function $P|Q$, the symbol <PQ> is equivalent to the disjunction of classical propositional connective $P \vee Q$, the symbol【P】 is equivalent to the necessary $*P$ and the symbol〖P〗is equivalent to the possible $+P$.

In the formal language L_{MP}, all propositional connectives and modal operators can be defined by a pair of parenthesis ( ), which is the only initial connective symbol.

The natural deduction system K of Modal Propositional Logic based on parenthesis notation can be built on the following rules:

Inference rule 01 P :P.

Inference rule 02 If Σ :P, then Σ, Δ :P.

Inference rule 11 If Σ, P, Q :R, and Σ, P, Q :『RR』, then Σ :『PQ』.

Inference rule 12 If Σ :『『PP』Q』, and Σ :Q, then Σ :P.

Inference rule 13 If Σ :『PQ』, and Σ : P, then Σ :『QQ』.

Inference rule 21 If Σ :【P『QQ』】, and Σ :【P】, then Σ :【Q】.

Inference rule 22 If : P, then :【P】.

In this system, only binary propositional connectives “『 』” and modal operator necessary “【 】” are included, which are defined by a unique initial triple connective “( ) ”.

4. The Origin and Development of Parenthesis Notation

The explicit introduction of the parenthesis notation was inspired by the Sheffer function and related work by Qingyu Zhang.

The Sheffer function reduces all propositional conjunctions to a single logical function, i.e. alternative denial or joint denial, usually denoted by the symbols “|” and “↓”, and using the logical functions, $P|Q$

can be expressed as $\neg P \vee \neg Q$, $P \downarrow Q$ can be expressed as $\neg P \wedge \neg Q$. Both alternative denial and joint denial are very expressive and can express all propositional conjunctions (Mendelson, 2015). Inspired by this, we consider whether it is possible to create more expressive logical constants that not only express all propositional conjunctions, but also further reduce other logical constants, such as quantifiers and modals, into one logical constant.

As early as 1995, Chinese logician Qingyu Zhang put forward: "In the common classical propositional formal system, there are always both connectives and parenthesis. That is to say, the connective and grouping functions required in the construction of well-formed formula are undertaken by two kinds of symbols respectively. In fact, these two functions can be undertaken by one kind of symbols in the classical propositional formal system." Therefore, he constructed a propositional formal language and its related axiom system, which only contains propositional variables, propositional constant "t" and parenthesis "( )", so as to show that parenthesis can also act as connectives. In his paper, Qingyu Zhang used the parenthesis "(PQ) " as a binary connective equivalent to the binary connective "$P \wedge \neg Q$" (Zhang, 1995).

In 1996, Qingyu Zhang established a first-order system without the common propositional connectives and quantifiers (Zhang, 1996). There are three kinds of initial signs in his first-order language:

(1) Tautology symbols and propositional symbols: T; $P_0, P_1, P_2, \ldots, P_n, \ldots$;

(2) Variables: $v_0, v_1, v_2, \ldots, v_n, \ldots$;

(3) Comma and parenthesis: , , (,).

In the 1996 paper, the parenthesis "(QR)" as a binary connective is equivalent to the binary connective "$Q \wedge \neg R$"; the symbol "(QxR) " as a variable connective is equivalent to the connective "$\exists x[Q \wedge \neg R]$". The common propositional connectives can be obtained by definition:

$\neg P := (\mathrm{T}P)$

$P \to Q := (\mathrm{T}(PQ))$

$P \wedge Q := (P(\mathrm{T}Q))$

$\exists xP := (\mathrm{T}x(\mathrm{T}P))$

$\forall xP := (\mathrm{T}(\mathrm{T}xP))$

In 2019, based on Qingyu Zhang's research, Guoping Du proposed the idea of using parenthesis to express all logical constants instead of using zero element connectives and he put forward parenthesis notation (Du, 2019). In 2020, parenthesis notation is extended from two-valued propositional logic and first-order quantifier logic to many-valued formal system (Du, 2020). In 2021, he applied the parenthesis notation in other axiomatic systems and extended it to arbitrary logical systems, and proved that only using a pair of parentheses can express all the logical constants in any formal system. The parenthesis notation as an independent notation is established.

The parenthesis notation differs from the infix notation and Polish notation. Since it is a new representation of logical constants proposed by Chinese scholars, we might also call it the Chinese notation.

References

Bonevac, D. (2012). "A History of Quantification." In *Handbook of the History of Logic, Volume 11: Logic*: *A History of Its Central Concepts* (Edited by Gabbay, Dov M., Pelletier, Francis Jeffry., & Woods, John., pp. 63-126). Amsterdam: Elsevier. https://doi.org/10.1016/B978-0-444-52937-4.50002-2

Du, G. P. (2019). First-Order Logic System based on Parenthesis Notation. *Journal of Anhui University* (*Philosophy and Social Sciences*), *43*, 35-41.

Du, G. P. (2019). A note on "logical systems without connectives". *Journal of Chongqing University of Technology (Social Science)*, *4*, 7-12.

Du, G. P. (2019). A "no...but..." -Type Natural Deduction Systems without Connectives. *Journal of Hunan University of Science & Technology (Social Science Edition)*, *3*, 21-24.

Du, G. P. (2020). A system of natural inference for Four-Valued Logic based on Parenthesis Notation, *Journal of Hubei University* (*Philosophy and Social Sciences*), *47*, 36-49.

Du, G. P. (2021). The effective proof the sustem of joint denial and its theorems. *Journal of Chongqing University of Technology (Social Science)*, *6*, 53-61.

Du, G. P. (2021). Natural deduction systems for "cannot get both" -Type propositional logic. *Journal of Guangxi University (Philosophy and Social Sciences)*, *3*, 52-56.

Mendelson, E. (2015). *Introduction to Mathematical Logic* (pp. 22-23). CRC Press. https://doi.org/10.1201/b18519

Łukasiewicz, J. (1966). *Elements of Mathematical Logic* (pp. 22-30). Oxford: Pergamon Press.

Zhang, Q. Y. (1995). Classical Propositional System without Connectives. *Philosophical Research*, *5*, 40-47.

Zhang, Q. Y. (1996). First Order System without Connectives and Quantifiers. *Philosophical Research*, *5*, 72-79.

Notes

Note 1. In this paper, square brackets [ ] are used to express the usual combination level and sequence relationship.

Note 2. In fact, there may be different definitions for the above definitions. For example, for all propositional connectives, universal quantifier and particular quantifier, parenthesis notation can have a more concise definition. Define a triple: $(PxQ)=_{def}\ \forall x[P[x]\to\neg Q[x]]$. For the formula (PxQ) which x does not occur in P and Q, we have $(PxQ) = \forall x[P[x]\to\neg Q[x]] = \forall x[P\to\neg Q] = \neg P\vee\neg Q$. Hence, (PxQ) can define all propositional connectives. In addition, $(\neg Qx\neg Q) = \forall x\ [\neg Q[x]\to\neg\neg Q[x]] = \forall x[\neg\neg Q[x]] = \forall x[Q[x]]$, hence, $(\neg Qx\neg Q)$ can define universal quantifier $\forall$.