Entailment as a Logical Basis for Deductive Reasoning

CHENG, Jingde

数理解析研究所講究録
第 709 巻 1989 年 199-220

Entailment as a Logical Basis for Deductive Reasoning

程 京徳
Jingde CHENG

Department of Computer Science and Communication Engineering
Kyushu University, Fukuoka 812, Japan

Abstract

This paper proposes to use the entailment that is a logical connective in entailment logic and relevance logics as a logical basis for deductive reasoning. The proposal is based on two ideas concerning the use of logic in computer science. One is that logic should be regarded as a description tool to represent an entailment relation between propositions. The other is that logic should be regarded as a reasoning tool to guarantee a logical validity of deductive reasoning in the sense of the entailment, i.e., the conclusion of a valid deductive reasoning should not be a tautological consequence but an entailment consequence for given premises. Using the entailment as a logical basis for deductive reasoning makes it possible for us to construct such logic systems where the validity of a conclusion of a deductive reasoning is dependent only on the validity of given premises and the correctness of the reasoning and independent of the concrete content of the conclusion.

Key words Entailment logic, Relevance logic,
Deductive reasoning, Validity of deductive reasoning

1. Introduction

In recent years, various deductive reasoning methods have extensively been used in many areas of computer science, both in theoretical researches and in practical applications, such as logic programming systems, deductive databases, expert systems, and knowledge-based systems. Many of them are fundamentally based on the classical mathematical logic where the modus ponens with the material implication is used as a logical basis for deductive reasoning. The modus ponens has such a form : from X and X→Y to infer Y where X and Y are logical formulas and "→" is the material implication.

Logic deals with what follows from what. It is the systematic study of the fundamental principles that underlie correct, necessary pieces of reasoning, as these occur in proofs, arguments, inferences, and deductions. The correctness of a piece of reasoning does not depend on what the reasoning is about so much as on how the reasoning is done; on the pattern of relationship between the various constituent ideas rather than on the actual ideas themselves. To get at the relevant aspects of such reasoning, logic must abstract its form from its content. A basic idea of logic is regarding a deductive reasoning form is valid if there is no reasoning of that form whose premisses are true and whose conclusion is false without regard for the concrete contents of the premisses and conclusion [Robinson-79].

The classical mathematical logic is the study of logic as a mathematical theory. Its function is to provide formal languages for describing the structures with which mathematicians work, and the methods of proof available to them. Mathematicians construct mathematical logic as a mathematical model of the systems to be studied, and then conduct what is essentially a pure mathematical investigation of the properties of this model. Therefore, An important feature of the classical mathematical logic is that it need not concern the nature of the real world in the sense that the world mathematicians study is the purely conceptual one of pure mathematics [Kleene-67, Barnes-75, Johnstone-87].

On the other hand, it is clear that computer science, both its theoretical aspect and its practical aspect, is not purely mathematical. According to ISO Standard 2382/1 (1984), computer science is "the branch of science and technology that is concerned with methods and techniques relating to data processing performed by automatic means". Today, the word "data processing" has included, in a broad sense, the meaning of processing "information" and "knowledge" which are closely concerned with the nature of our real world. Any problem or project on which computer scientists or engineers work is certainly concerned with processing information and/or knowledge. Therefore, we may and should require that use of logic in computer science, especially in deductive reasoning, is consistent with our ordinary logical thinking. In fact, logic is extensively used in computer science precisely because it is ordinarily used by us with a natural language form just as a basic tool in presentation and analysis of arguments.

However, from the viewpoint of our ordinary logical thinking, there is a theoretical problem in using the material implication as a logical basis for deductive reasoning. That is, some formulas, such as "$X \rightarrow (Y \rightarrow X)$" and "$\neg X \rightarrow (X \rightarrow Y)$" which may be axioms or provable formal theorems in the classical mathematical logic, are regarded as "implicational paradoxes" in the sense of our ordinary logical thinking. As a result, for a conclusion of such a deductive reasoning, we cannot directly accept it as a "valid" conclusion in the sense of our ordinary logical thinking, even if each of given premises is "valid" in the sense of our ordinary logical thinking. In order to evaluate whether the conclusion is valid, we have to investigate the concrete content of the conclusion.

We consider the use of logic in computer science should be based on such a logical basis as is consistent with our ordinary logical thinking. A basic requirement for a software system working with a deductive reasoning mechanism should be that it is possible for users to directly accept a deductively reasoned conclusion as a "valid" conclusion in the sense of our ordinary logical thinking if each of given premises is "valid". This paper proposes to use the entailment that is a logical connective in entailment logic and relevance logics as a logical basis for deductive reasoning. The proposal is based on two ideas concerning the use of logic in computer science. One is that logic should be regarded as a description tool to represent an entailment relation between propositions. The other is that logic should be regarded as a reasoning tool to guarantee a logical validity of deductive reasoning in the sense of the entailment, i.e., the conclusion of a valid deductive reasoning should not be a tautological consequence but should be an entailment consequence for given premises.

In Section 2, we point out some problems in using the material implication as a logical basis for deductive reasoning. In Section 3, we review briefly relevance logics [Anderson-75] and entailment logic [Lin-85a,b]. In Section 4, we summarize propositional aspect of entailment logic. In Section 5, we present a formal semantics for a subclass of propositional entailment logic. In Section 6, we discuss the problem of validity of deductive reasoning. Concluding remarks are given in Section 7.

2. Problems of Using the Material Implication as a Logical Basis for Deductive Reasoning

In our ordinary logical thinking, the notion of entailment, which is often informally said "entails" or "if ··· then", plays a central role. The entailment relation between two propositions, for instance, "X entails Y" or "if X then Y", depends not only on the truth of X and Y but also more essentially on a necessarily relevant relation between X and Y. On the other hand, in the classical mathematical logic, the material implication relation between two propositions depends only on the truth of X and Y and is independent of any necessarily relevant relation between X and Y. There is no problem if one use the material implication accurately and strictly according to its formal truth functional interpretation, as mathematical logicians do. But unfortunately, many people often use the material implication with an informal interpretation of the entailment notion, e.g., "if ··· then", and even some texts suggest such use to readers. This brings about the problem of implicational paradoxes.

For example, $p \rightarrow (q \rightarrow p)$, which is generally as an axiom in the classical mathematical logic, is equivalent to $q \rightarrow (\neg p \vee p))$. In terms of the logic, this means that "a identically true proposition is implied by any proposition". $\neg p \rightarrow (p \rightarrow q)$, which is a provable formal theorem in the logic, is equivalent to $(\neg p \wedge p) \rightarrow q$. This means that "a identically false proposition implies any proposition". However, in the sense of our ordinary logical thinking, we can accept neither proposition "if p then t for a identically true proposition t and any proposition p" nor proposition "if f then p for a identically false proposition f and any proposition p" as a "valid" proposition.

Due to the problem of implicational paradoxes, when using the material implication as a logical basis for deductive reasoning, we may confront with the following dilemma. On the one hand, if we use the material implication accurately and strictly according to its formal truth functional interpretation, then a material implication proposition, e.g., "if snow is black then $3+4=7$" or "if snow is white then $3+4=7$", each of which is "invalid" in the sense of our ordinary logical thinking, can be accepted as a valid proposition used in a deductive reasoning. On the other hand, if we use the material implication with an "if … then" interpretation of the entailment, then we cannot expect to obtain a "valid" conclusion with respect to entailment between two propositions, e.g., "if X then Y", from a deductive reasoning even if each of given premises is "valid" in the sense of our ordinary logical thinking. The classical mathematical logic does not provide such a guarantee.

Therefore, in the framework of the classical mathematical logic, for a deductively reasoned conclusion, we cannot directly accept it as a valid conclusion in the sense of our ordinary logical thinking even if each of given premises is "valid" in the sense of our ordinary logical thinking. In order to evaluate whether the conclusion is valid, we have to investigate the concrete content of the conclusion. This is inconsistent with the basic idea of logic that the validity of a conclusion of a reasoning should be dependent only on the validity of given premises and the correctness of the reasoning and independent of the concrete content of the conclusion. If the validity of a conclusion reasoned by a software system has to be evaluated based on the investigation of the concrete content of the conclusion, then this is just as the correctness of codes generated by a compiler has to be guaranteed by checking the codes by users themselves !

The cause of the problem of implicational paradoxes is that the material implication notion is intrinsically different from the entailment notion of human ordinary logical thinking in semantics. Therefore, there is a key question. Can we have a logical connective whose meaning is consistent with the entailment notion of human ordinary logical thinking and has a formal interpretation? In order to construct more "natural" and/or "logical" deductive reasoning mechanism, we have to answer this problem.

3. Relevance logics and Entailment Logic

An obvious strategy for solving the problem of implicational paradoxes in the classical mathematical logic is to formalize the notion of "entailment" as a relation stronger than the notion of "material implication" such that all implicational paradoxes are unprovable in an axiom system based on the formalization. Based on this idea, a number of "paradox-free" logics have been proposed. However, "the simple avoidance of paradox is hardly sufficient to characterize what we mean by entailment, and most of the formalized proposals based on this single aim have been so artificial as to make it virtually impossible to get a clear grasp of their formal properties" [Anderson-60].

The first interesting proposal for "paradox-free" logics is Ackermann's logic system which provably avoids the implicational paradoxes [Ackermann-56]. Ackermann introduced a new logical connective called "strenge implikation(rigorous implication)", which provides a strong and natural sort of implication relation, and construct a calculus Π' of "strenge implikation". Unlike the previous proposals, "Ackermann's system, which has the required property, seems intuitively natural, and strong enough to be of interest" [Anderson-57]. However, Ackermann has not given a semantical definition for his rigorous implication and a formal model for his calculus Π' of "strenge implikation". Anderson and Belnap modified and reconstructed Ackermann's system into an equivalent logic system, called "system E of entailment" [Anderson-58,60,75]. They also interested E's some neighboring logic systems which are obtained by adding (dropping) some axioms into (from) system E [Anderson-75]. These logic systems are generally called "relevance logics" [Anderson-75, Morgan-76, Routley-84, Thistlewaite-88].

Lin's "entailment logic" [Lin-85a,b] is another interesting proposal which is independent of the work of Ackermann et al. Lin investigated logical meaning of the sufficient conditional relation in human ordinary logical thinking and introduced a new logical connective, called "entailment", as a logical abstraction of the sufficient conditional relation. He proposed two new logical concepts, i.e., the first independence in the entailment relation between two propositions and the second independence in a deductive inference. He also constructed a propositional calculus, denoted by *Cm*, which provably avoids the implicational paradoxes, and a notional calculus, denoted by *Cn*, with *Cm* as the underlying propositional logic. However, Lin has not given a formal model for his entailment logic system.

The above studies by philosophers and/or logicians are presented in the philosophical literature. However, such studies has received little attention from computer scientists [Morgan-76, Thistlewaite-88].

We compare the main deference between the relevance logics and the entailment logic in their three aspects, i.e., claim, syntax, and semantics. Both constructors of the relevance logics and the entailment logic claim that one should take the heart of logic to lie in the notion of entailment [Anderson-75, Lin-85a]. But, the constructors of relevance logics regard the relevance logics as a branch of mathematical logic [Anderson-75] and the constructor of entailment logic rejects mathematical logic and regard the entailment logic as a modern development of classical formal logic [Lin-85a]. Therefore, the approach of Ackermann et al is "syntactical" or "proof-theoretical", i.e., they investigate syntactic features of the implicational paradoxes and constructs formal axiom systems which provably avoids the paradoxes at first and then investigate formal semantics for the axiom systems. On the other hand, Lin's approach is "semi-semantical" or "semi-model-theoretical", i.e., he investigate logical meaning of the notion of entailment in human ordinary logical thinking and introduced a new logical connective as a logical abstraction of the entailment at first and then construct the propositional calculus and the notional calculus for the entailment. From a syntactical view point, a propositional entailment logic can be regarded as a relevance logic because it can be obtained by dropping some axioms from system E and adding some axioms. A complete entailment logic is a notional logic and includes no quantifiers. But there are no notional calculus of the relevance logics. The main difference between the relevance logics and the entailment logic in semantics is that Lin first explicitly introduced the concepts of the first independence and the second independence in constructing entailment logic. Based on these concepts, Lin successfully solved the failure problem of disjunctive syllogism [Hughes-72, Thistlewaite-88, Lin-85a].

We consider the concepts of the first independence and the second independence are important for discussing the validity of a deductive reasoning. Therefore, below we will investigate what is a valid deductive reasoning based on Lin's entailment logic.

4. Propositional Entailment Logic

A formal axiom system for the propositional entailment logic, denoted by *Cm* [Lin-85a,b], is summarized as follows, where X, Y, and Z are syntactical variables.

Alphabet :

(1) a denumerable set of proposition variable symbols;

(2) the logical connective symbols $\neg$(negation), $\wedge$(conjunction), and $\Rightarrow$(entailment);

(3) brackets (and).

Formulas :

(1) every proposition variable, also called *atomic formula*, is a formula;

(2) if X and Y are formulas then $\neg X$, $(X \wedge Y)$, $(X \Rightarrow Y)$ are formulas, where the outermost brackets of a formula can be omitted;

(3) only those defined by (1) and (2) are formulas.

Axiom Schemata :

A_1 $X \Rightarrow \neg\neg X$

A_2 $(X \Rightarrow (Y \Rightarrow Z)) \Rightarrow (Y \Rightarrow (X \Rightarrow Z))$

A_3 $(Y \Rightarrow Z) \Rightarrow ((X \Rightarrow Y) \Rightarrow (X \Rightarrow Z))$

A_4 $(\neg X \Rightarrow Y) \Rightarrow (\neg Y \Rightarrow X)$

A_5 $(X \wedge \neg Y) \Rightarrow \neg(X \Rightarrow Y)$

A_6 $X \Rightarrow (X \wedge X)$

A_7 $(X \wedge Y) \Rightarrow X$

A_8 $(X \wedge Y) \Rightarrow (Y \wedge X)$

A_9 $((X \Rightarrow Y) \wedge (X \Rightarrow Z)) \Rightarrow (X \Rightarrow (Y \wedge Z))$

A_{10} $((X \wedge \neg(\neg Y \wedge \neg Z)) \Rightarrow \neg(\neg(X \wedge Y) \wedge \neg(X \wedge Z)))$

Inference Rules :

R_1 From X and $X \Rightarrow Y$ to infer Y

R_2 From X and Y to infer $X \wedge Y$

The meanings of logical connectives of the propositional entailment logic are as follows.

$\neg X$:	"X does not hold"
$X \wedge Y$:	"Both X and Y hold"
$X \Rightarrow Y$:	"It can be determined that there is no such case as X holds and Y does not hold with neither determining whether X holds or not nor determining whether Y holds or not"

The entailment "⇒" requires that every entailment formula satisfies the following two conditions in semantics :

(1) there is no such case as the antecedent is true and the consequent is false;
(2) condition (1) can be determined with neither determining whether antecedent is true or false nor determining consequent is true or false.

The above condition (2) is called *first independence* [Lin-85a,b]. The most intrinsic difference between the entailment and the material implication is that the former has the first independence which is a requirement for a necessary relevant relation between two propositions but the latter does not have such requirement.

There may be two possible extensions with *Cm* as the underlying propositional logic. One is to extend *Cm* into a notional logic [Lin-85a,b]. The other is to extend *Cm* into a first order logic.

Now, we give some definitions for future discussion.

Definition 4.1 We call a formula a *classical formula* if no entailment connective occurs in it. We use C_c to denote the set of all classical formulas. We call a formula with form $X \Rightarrow Y$ an *entailment formula*, where X is called the *antecedent* of this entailment formula and Y is called the *consequent*. □

We inductively define the *degree* of an entailment connective.

Definition 4.2 We say the degree of the entailment connective in an entailment formula $X \Rightarrow Y$ is 1 if both X and Y are classical formulas. We say the degree of the entailment connective in an entailment formula $X \Rightarrow Y$ is $k+1$ if there exists a positive integer k such that k is the highest degree of entailment connectives in X and Y. We call a formula a *first degree entailment formula* if the highest degree of its entailment connectives is 1. We use C_f to denote the set of all first degree entailment formulas. We call a formula a *second degree entailment formula* if the highest degree of its entailment connectives is 2. We use C_s to denote the set of all second degree entailment formulas. □

A classical formula is also called a <u>*zero degree entailment formula*</u>.

In the following discussion, we use P to denote a set of formula $\{Z_1, Z_2, \cdots, Z_n\}$ and use q to denote a single formula.

Definition 4.3 A <u>*deduction of q from P*</u> is a finite sequence of formulas such that each member of the sequence is either an axiom, a member of P, or is obtained by using any of the inference rules from two earlier members of the sequence, and the last member of the sequence is q. We call all members of P the <u>*premisses*</u> and q the <u>*consequence*</u> of the deduction. A deduction q from the empty set of premisses is called a <u>*proof*</u> and q is called a <u>*formal theorem*</u>. □

Definition 4.4 We say P <u>*syntactically entails*</u> q or q <u>*is provable from*</u> P, write $P \vdash q$, if and only if q satisfies any of the following conditions :

(1) q is an axiom;
(2) $q \in P$;
(3) there exist some r and t such that $P \vdash r$, $P \vdash t$ and $q = r \wedge t$; or
(4) there exists some t such that $P \vdash t$ and $P \vdash (t \Rightarrow q)$. □

Clearly, $P \vdash q$ if and only if there exists a deduction of q from P, and $\Phi \vdash q$ if and only if q is a formal theorem, where Φ denotes the empty set.

<u>5. Formal Semantics of Second Degree Propositional Entailment Logic</u>

It is not a completely solved problem to provide an adequate formal model for the entailment logic. We are investigating an algebra semantics for full propositional entailment logic. Below, we give a formal semantics for a subclass of the entailment logic, named <u>*second degree propositional entailment logic*</u> and denoted by PEL_{sd}.

The alphabet of PEL_{sd} are the same as that of Cm. The set of all formulas of PEL_{sd} is $C_c \cup C_f \cup C_s$. The axiom schemata of PEL_{sd} are A_1, and $A_4 \sim A_{10}$ of Cm. The inference rules of PEL_{sd} are the same as that of Cm.

First of all, we describe some basic concepts of lattice theory [Birkhoff-61, Szasz-63, Rutherford-65].

Definition 5.1 A *partial order* on a set A is a binary relation relation $\leqq \subseteq A \times A$ satisfying the following conditions :

Reflexivity : For every a in A, $a \leqq a$;

Antisymmety : For every a, a' in A, if both $a \leqq a'$ and $a' \leqq a$ hold, then $a = a'$;

Transitivity : For every a, a', a" in A, if both $a \leqq a'$ and $a' \leqq a''$ hold, then $a \leqq a''$.

If $\leqq$ is a partial order on A, we call the pair (A, $\leqq$) a *poset* (short for *partially ordered set*). □

Definition 5.2 Let (A, $\leqq$) be a poset. Let A' be a subset of A. An element b of A is called an *lower bound* (l.b.) for A', if $b \leqq a$ for every a in A'; b is called *greatest lower bound* (g.l.b.) for A', if $b' \leqq b$ for every l.b. b' for A'. An element b of A is called an *upper bound* (u.p.) for A' if $a \leqq b$ for every a in A', b is called *least upper bound* (l.u.p.) for A' if $b \leqq b'$ for every u.b. b' for A'. We denote g.l.b. and l.u.p. for A' by $\cap$ A' and $\cup$ A' respectively. If A' has only two elements, we write $a_1 \cap a_2$ (read : a_1 meet a_2) and $a_1 \cup a_2$ (read : a_1 join a_2), respectively, for $\cap \{a_1, a_2\}$ and $\cup \{a_1, a_2\}$. □

Definition 5.3 A *meet-semilattice* is a poset (L, $\leqq$) such that any two its elements have a g.l.b.; a *join-semilattice* is a poset (L, $\leqq$) such that any two its elements have a l.u.b.; a *lattice* is a poset (L, $\leqq$) which is both a meet-semilattice and a join-semilattice; a *complete lattice* is a poset (L, $\leqq$) in which every subset has both a g.l.b. and a l.u.b.. □

Now, we define a lattice for giving formal semantics of PEL_{sd}.

Definition 5.4 An *entailment lattice* is a quadruplet <L, $\leqq$, N, T> that satisfies the following conditions :

(1) (L, $\leqq$) is a meet-semilattice;

(2) N is a unary function, which satisfies the following conditions, on L :
<1> for all a ∈ L, N(N(a)) = a, and
<2> for all a, b ∈ L, if a ≦ b then N(b) ≦ N(a);

(3) T is a sublattice, which satisfies the following conditions, of L :
<1> for all a ∈ L, N(a) ∈ T iff a ∉ T,
<2> for all a, b ∈ L, a ∩ b ∈ T iff a ∈ T and b ∈ T, and
<3> there exists no a, b ∈ L such that a ∈ T, N(b) ∈ T, and a ≦ b. □

We use F and T, called truth values, to represent "false" and "true" respectively. Let D be the set of all atomic formulas. Let C_c be the set of all classical formulas defined on D, C_f the set of all first degree entailment formulas defined on D, and C_s the set of all second degree entailment formulas defined on D, respectively.

Definition 5.5 An *interpretation* of PEL_{sd} is a quadruplet I = <L_e, v_f, v_s, s> that satisfies the following conditions :

(1) L_e = <L, ≦, N, T> is an entailment lattice;

(2) v_f is a mapping, v_f : D → L;

(3) v_s is a mapping, v_s : { X⇒Y | X, Y ∈ C_c } → L, which satisfies the following conditions :
<1> v_s(X⇒Y) ∈ T iff s(X) ≦ s(Y),
<2> v_s(¬X⇒Y) ≦ v_s(¬Y⇒X),
<3> s(X∧¬Y) ≦ N(v_s(X⇒Y)),
<4> (v_s(X⇒Y) ∩ v_s(X⇒Z)) ≦ v_s(X⇒(Y∧Z)),
<5> if s(X) ≦ v_s(Y⇒Z) then s(X∧Y) ≦ v_s(Z);

(4) s is a mapping, s : C_c ∪ C_f → L, which satisfies the following conditions :
<1> s(d) = v_f(d) for all d ∈ D,
<2> s(d) = v_s(d) for all d ∈ { X⇒Y | X, Y ∈ C_c },
<3> s(¬X) = N(s(X)) for all X ∈ C_c ∪ C_f,
<4> s(X∧Y) = s(X) ∩ s(Y) for all X, Y ∈ C_c ∪ C_f. □

Definition 5.6 For every interpretation I = <L_e, v_f, v_s, s> and every formula f of PEL_{sd}, the truth value of f, denoted by I(f), is inductively defined as follows, where X, Y ∈ C_c ∪ C_f ∪ C_s and M, N ∈ C_c ∪ C_f.

(1) For every $X \in \mathbf{D}$, $I(X) = T$ iff $v_f(X) \in \mathbf{T}$, and otherwise $I(X) = F$.
(2) $I(\neg X) = T$ iff $I(X) = F$, and otherwise $I(\neg X) = F$.
(3) $I(X \wedge Y) = T$ iff $I(X) = T$ and $I(Y) = T$, and otherwise $I(X \wedge Y) = F$.
(4) $I(M \Rightarrow N) = T$ iff $s(M) \leqq s(N)$, and otherwise $I(M \Rightarrow N) = F$. □

In the following discussion, we use P to denote a set of formula $\{Z_1, Z_2, \cdots, Z_n\}$ and use q to denote a single formula.

Definition 5.7 We say P *semantically entails* q, write $P \models q$, if and only if $I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow q) = T$ for any interpretation $I = <L_e, v_f, v_s, s>$. We say a formula q is *valid*, write $\models q$, if and only if $I(q) = T$ for any interpretation $I = <L_e, v_f, v_s, s>$. □

From this definition, we have the following Theorem 5.1 as a direct result.

Theorem 5.1 $\{X_1, X_2, \cdots, X_n\} \models Y$ iff $\models (X_1 \wedge X_2 \wedge \cdots \wedge X_n) \Rightarrow Y$. □

Corollary $X \models Y$ iff $\models X \Rightarrow Y$. □

The following Theorem 5.2 shows that PEL_{sd} has a model defined based on an entailment lattice.

Theorem 5.2 All axioms of PEL_{sd} are valid.

Proof This is easy to prove by applying Definition 5.6 to all axiom schemata of PEL_{sd}. □

The following two theorems shows that the two inference rules of the propositional entailment logic are valid.

Theorem 5.3 If $P \models X$ and $P \models X \Rightarrow Y$ then $P \models Y$.

Proof Suppose $P \models X$ and $P \models X \Rightarrow Y$. Then

$$I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow X) = T \text{ for any interpretation I, and}$$
$$I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow (X \Rightarrow Y)) = T \text{ for any interpretation I.}$$

According to Definition 5.5, if $I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow (X \Rightarrow Y)) = T$ then $I(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n \wedge X \Rightarrow Y) = T$. Therefore, for any interpretation $I = <L_e, v_f, v_s, s>$, the following formulas hold.

$$s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \leqq s(X) \text{ and } s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n \wedge X) \leqq s(Y)$$

According to Definition 5.5, $s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n \wedge X) = s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \cap s(X)$. Moreover, according to lattice theory, if $a \leqq b$ then $a \cap b = a$. Therefore, we have

$$s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \leqq s(Y)$$

Consequently,

$$I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow Y) = T \text{ for any interpretation I, i.e., } P \models Y. \quad \Box$$

Corollary If $\models X$ and $\models X \Rightarrow Y$ then $\models Y$.

Proof Suppose $\models X$ and $\models X \Rightarrow Y$. Then

$$I(X) = T \text{ for any interpretation I, and}$$
$$I(X \Rightarrow Y) = T \text{ for any interpretation I.}$$

Therefore, for any interpretation $I = <L_e, v_f, v_s, s>$, the following formulas hold.

$$s(X) \in \mathbf{T} \text{ and } s(X) \leqq s(Y)$$

So, according to the properties of entailment lattices (see Definition 5.4), there must be $s(Y) \in \mathbf{T}$, i.e., $\models Y$. $\Box$

Theorem 5.4 If $P \models X$ and $P \models Y$ then $P \models X \wedge Y$.

Proof Suppose $P \models X$ and $P \models Y$. Then

$$I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow X) = T \text{ for any interpretation I, and}$$
$$I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow Y) = T \text{ for any interpretation I.}$$

Therefore, for any interpretation $I = <L_e, v_f, v_s, s>$, the following formulas hold.

$$s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \leqq s(X) \text{ and } s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \leqq s(Y)$$

So, according to lattice theory, the following formula holds.

$$s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \leqq s(X) \cap s(Y)$$

Moreover, according to Definition 5.5, $s(X \wedge Y) = s(X) \cap s(Y)$. Therefore, we have

$$s(Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \leqq s(X \wedge Y).$$

Consequently,

$I((Z_1 \wedge Z_2 \wedge \cdots \wedge Z_n) \Rightarrow (X \wedge Y)) = T$ for any interpretation I, i.e., $P \models X \wedge Y$. □

Corollary If $\models X$ and $\models Y$ then $\models X \wedge Y$. □

From Theorems 5.1 ~ 5.4, we have the following

Theorem 5.5 (The Soundness Theorem of PEL_{sd}) If $P \vdash q$ then $P \models q$; if $\vdash q$ then $\models q$. □

We may say, similar to the classical mathematical logic, that a formula X <u>*follows from*</u> a set P of formulas, write $P \models_f X$, if and only if $I(X)=T$ for every interpretation with $I(Y)=T$ for every $Y \in P$. But, note that in the terms of the propositional entailment logic, "a formula X follows from a set P of formulas" does not mean "P semantically entails X".

In the classical mathematical logic, a fundamental fact is $\{X_1, X_2, \cdots, X_n\} \models_f Y$ iff $\models_f (X_1 \wedge X_2 \wedge \cdots \wedge X_n) \rightarrow Y$ or $X \models_f Y$ iff $\models_f X \rightarrow Y$. However, in the propositional entailment logic, we only have the following Theorem 5.5 but do not have its converse theorem. This shows one of the differences between the entailment and the material implication in semantics.

Theorem 5.6 If $\models_f (X_1 \wedge X_2 \wedge \cdots \wedge X_n) \Rightarrow Y$ then $\{X_1, X_2, \cdots, X_n\} \models_f Y$; if $\models_f X \Rightarrow Y$ then $X \models_f Y$. □

<u>6. Valid Deductive Reasoning</u>

In the framework of the classical mathematical logic, from any given proposition X and the axiom $X \rightarrow (Y \rightarrow X)$, we can infer $Y \rightarrow X$ by using the modus ponens with the material implication. It is clear that the inference is not necessarily regarded to be valid from the viewpoint of human ordinary logical thinking because there may be no necessarily relevant relation between Y and X. Then, what is a "valid" deductive reasoning? Let us consider the question. This is an old and important problem both in philosophy and in logical science. For computer science, when we intend to develop a software system working with a deductive reasoning mechanism, such as a logic programming system, deductive database, expert system, and knowledge-based system, to consider and answer the above problem is very important for designing the deductive reasoning mechanism and correctly evaluating conclusions deductively reasoned by the software system.

Definition 6.1 An entailment formula $X \Rightarrow Y$ is called a *valid entailment formula* if $\models X \Rightarrow Y$. A valid entailment formula $X \Rightarrow Y$ is called a *derivative formula* if $\models X$ can be determined only after $\models Y$ has been determined, i.e., if not $\models Y$ then not $\models X$. A valid entailment formula $X \Rightarrow Y$ is called a *deductive formula* if $\models X$ without determining $\models Y$, i.e., $\models X$ whether $\models Y$ or not. □

The property that $\models X$ can be determined without determining $\models Y$ is called the *second independence* [Lin-85a,b].

For example, $(X \wedge Y) \Rightarrow Y$ is a derivative formula and $(X \wedge (X \Rightarrow Y)) \Rightarrow Y$ is a deductive formula.

Theorem 6.1 If $X \Rightarrow Y$ is a valid entailment formula, then X and Y must share a propositional variable.

Proof Suppose $X \Rightarrow Y$ is a valid entailment formula, i.e., $\models X \Rightarrow Y$. If no propositional variable occurs both in X and in Y, then we can construct an interpretation $I = \langle L_e, v_f, v_s, s \rangle$ such that $s(Y) \leqq s(X)$. But this is inconsistent with the fact $\models X \Rightarrow Y$. Therefore, X and Y must share a propositional variable. □

Corollary If $X \Rightarrow Y$ is a valid entailment formula and a first degree entailment formula, then X and Y must share all propositional variables occurring in Y. □

Theorem 6.2 No first degree entailment formula is a deductive formula.

Proof Suppose $X \Rightarrow Y$ is a first degree entailment formula, then by the corollary of Theorem 6.1 X and Y must share all propositional variables occurring in Y. Therefore, if not $\models Y$ then not $\models X$, i.e., $X \Rightarrow Y$ is not a deductive formula. □

Theorem 6.2 means that "every tautological implicational formula is a correct inference form" is incorrect in terms of the propositional entailment logic.

An interesting fact is that every "logical derivative law" specified in formal logic texts can be regarded a first degree entailment formula and every "logical deductive law" specified in formal logic texts can be regarded a second degree deductive formula.

In the following discussion, we use the following abbreviations (defined logical operators).

- $X \vee Y$: $\neg(\neg X \wedge \neg Y)$
- $X \rightarrow Y$: $\neg(X \wedge \neg Y)$
- $X \leftrightarrow Y$: $(X \rightarrow Y) \wedge (Y \rightarrow X)$
- $X \not> Y$: $\neg X \Rightarrow Y$
- $X ! Y$: $\neg(X \Rightarrow \neg Y)$
- $X \Leftrightarrow Y$: $(X \Rightarrow Y) \wedge (Y \Rightarrow X)$

For notional simplicity, we may define a priority order of the primitive and defined logical operators : $\neg$, $\wedge$, $\vee$, $!$, $\not>$, $\rightarrow$, $\leftrightarrow$, $\Rightarrow$, $\Leftrightarrow$, where the priority order of a left operator is higher that of the right one.

The derivative laws used in our ordinary logical thinking and discussed in formal logic are given by the notation of entailment logic as follows [Lin-85a].

- $X \Leftrightarrow X$
- $X \wedge X \Leftrightarrow X$
- $X \vee X \Leftrightarrow X$
- $\neg\neg X \Leftrightarrow X$
- $X \wedge Y \Leftrightarrow Y \wedge X$
- $X \vee Y \Leftrightarrow Y \vee X$
- $(X \wedge Y) \wedge Z \Leftrightarrow X \wedge (Y \wedge Z)$
- $(X \vee Y) \vee Z \Leftrightarrow X \vee (Y \vee Z)$
- $X \wedge (Y \vee Z) \Leftrightarrow (X \wedge Y) \vee (X \wedge Z)$
- $X \vee (Y \wedge Z) \Leftrightarrow (X \vee Y) \wedge (X \vee Z)$
- $\neg(X \wedge Y) \Leftrightarrow \neg X \vee \neg Y$
- $\neg(X \vee Y) \Leftrightarrow \neg X \wedge \neg Y$
- $X \wedge Y \Rightarrow X$
- $X \Rightarrow X \vee Y$

The deductive laws used in our ordinary logical thinking and discussed in formal logic are given by the notation of entailment logic as follows [Lin-85a].

- $(X \wedge (X \Rightarrow Y)) \Rightarrow Y$
- $(\neg Y \wedge (X \Rightarrow Y)) \Rightarrow \neg X$
- $(\neg X \wedge (\neg X \Rightarrow Y)) \Rightarrow Y$
- $(X \wedge (\neg X \Rightarrow Y) \wedge (Y \Rightarrow \neg X)) \Rightarrow \neg Y$
- $(X \wedge (X \Rightarrow \neg Y)) \Rightarrow \neg Y$
- $((X \Rightarrow Z) \wedge (Y \Rightarrow Z) \wedge (X \vee Y)) \Rightarrow Z$
- $((X \Rightarrow M) \wedge (Y \Rightarrow N) \wedge (X \vee Y)) \Rightarrow (M \vee N)$
- $((X \Rightarrow Y) \wedge (X \Rightarrow Z) \wedge (\neg Y \vee \neg Z)) \Rightarrow \neg X$
- $((X \Rightarrow M) \wedge (Y \Rightarrow N) \wedge (\neg M \vee \neg N)) \Rightarrow (\neg X \vee \neg Y)$
- $(X \Rightarrow Y) \Rightarrow (\neg Y \Rightarrow \neg X)$
- $(X \Rightarrow (Y \wedge \neg Y)) \Rightarrow \neg X$
- $(X \Rightarrow \neg X) \Rightarrow \neg X$
- $((X \Rightarrow Y) \wedge (Y \Rightarrow Z)) \Rightarrow (X \Rightarrow Z)$
- $((X \Rightarrow M) \wedge (Y \Rightarrow N)) \Rightarrow ((X \wedge Y) \Rightarrow (M \wedge N))$
- $((X \Rightarrow M) \wedge (Y \Rightarrow N)) \Rightarrow ((\neg M \wedge \neg N) \Rightarrow (\neg X \wedge \neg Y))$

Note that all above deductive lows are second degree deductive formulas. This fact means that the second degree entailment formulas are sufficient for our ordinary logical thinking.

In the framework of the propositional entailment logic, a <u>*deductive reasoning*</u> for a formula X from a given set P of formulas, called the <u>*premises*</u> of the deductive reasoning, can be regarded as $P \vdash X$.

Definition 6.2 We say <u>*a deductive reasoning is valid*</u> if and only if it satisfies the following two conditions :

(1) the modus ponens "from X and $X \Rightarrow Y$ to infer Y" is applied at least once;
(2) there exists at least one entailment formula $f \in P$ which is a deductive formula. □

For example, "by R2 from X, Y to infer $X \wedge Y$" and "by R1 from $X \wedge Y$, $(X \wedge Y) \Rightarrow Y$ to infer Y" are not valid deductive reasoning. "by R1 from $X \wedge (X \Rightarrow Y)$, $(X \wedge (X \Rightarrow Y)) \Rightarrow Y$ to infer Y" is a valid deductive reasoning.

Thus, we have the following proposition.

Proposition The conclusion of a valid deductive reasoning is not a tautological consequence but an entailment consequence for given premises. □

7. Concluding Remarks

We have presented why it is not suitable to use the material implication as a logical basis for deductive reasoning and have discussed what is a valid deductive reasoning based on the entailment notion that is a logical connective in entailment logic and relevance logics. The discussion leads us to proposes to use the entailment as a logical basis for deductive reasoning.

Now, let us discuss what contributions to computer science it can make to use the entailment as a logical basis for deductive reasoning.

First, the entailment notion provides a logical basis for more clearly and naturally describing "if ··· then", "only ··· if", and "if and only if" in knowledge representation.

Second, we can reject ambiguous descriptions for some central concepts of declarative semantics of logic programs, e.g., "logical consequence", "correct answer", and so on [Lloyd-87], and then give more clear and "logical" descriptions for such concepts using the entailment notion.

Third, using the entailment as a logical basis for deductive reasoning makes it possible for us to construct such logic systems where the validity of a conclusion of a deductive reasoning is dependent only on the validity of given premises and the correctness of the reasoning and independent of the concrete content of the conclusion. As a result, a logic programming system or deductive database based on the entailment logic may answer such a query as "is judgment 'if X then Y' correct?".

Finally, for given premises, i.e., knowledge, a theorem prover based on the entailment logic can deduce many new entailment formulas as valid conditions with respect to the entailment relation between two propositions, i.e., inference rules or new knowledge. This provides a logical basis for automatic synthesis of program, automatic generation of production rules of a production system, maintenance of a knowledge-based system.

Some important problems related to practical applications of the entailment logic are :

(1) Can we provide an adequate formal semantics for the full entailment logic with which we can discuss the soundness and completeness of the entailment logic?

(2) Is the entailment logic decidable for provability and validity?

(3) What automatic theorem proving method is suitable for the entailment logic?

Another further work is to compare the entailment logic and the relevance logics with other non-classical logic systems such as Nute's "conditional logic" [Nute-80], McDermott's Nonmonotonic Logic [McDermott-80,82], and so on.

The proposal to use the entailment as a logical basis for deductive reasoning is one of the results of our investigation for constructing a suitable logic system for deductive reasoning based concurrent program debugging [Cheng-88]. We developed an execution monitor for concurrent Ada programs that can record the execution history of a monitored program [Cheng-87]. In order to provide a powerful means for its users to debug a concurrent Ada program, we intend to develop a deductive reasoning based concurrent program debugging approach [Cheng-88]. In the approach, debugging a program can be regarded as queries and updates on a deductive database which contains the program specification, program source text, execution histories, and knowledge about program errors; detecting a program error can be regarded as a deductive proof for a formal theorem which is a logical formula specifying the error.

We are working for extending the entailment logic into a temporal entailment logic in order to construct a suitable logic system for deductive reasoning based concurrent program debugging [Cheng-88].

Acknowledgements

The author would like to thank Prof. Ushijima for his continuing guidance and valuable comments. The author also wish to thank Mr. Zhou for helpful discussions.

References

[Ackermann-56] Ackermann, W. : Begründung einer strengen Implikation, The Journal of Symbolic Logic, Vol. 21, pp. 113-128, 1956.

[Anderson-57] Anderson, A. R. : Review of [Ackermann-56], The Journal of Symbolic Logic, Vol. 22, pp. 327-328, 1957.

[Anderson-58] Anderson, A. R. and Belnap, N. D. Jr. : A Modification of Ackermann's 'rigorous implication', The Journal of Symbolic Logic, Vol. 23, pp. 457-458, 1958.

[Anderson-60] Anderson, A. R. : Completeness Theorems for the System E of Entailment and EQ of Entailment with Quantification, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, Bd. 6, S. 201-216, 1960.

[Anderson-75] Anderson, A. R. and Belnap, N. D. Jr. : Entailment : The Logic of Relevance and Necessity, Vol. 1, Princeton University Press, 1975.

[Barnes-75] Barnes, D. W. and Mack, J. M. : An Algebraic Introduction to Mathematical Logic, Springer-Verlag, 1975.

[Birkhoff-61] Birkhoff, G. : Lattice Theory, Revised Edition, American Mathematical Society, 1961.

[Cheng-87] Cheng, J., Araki, K., and Ushijima, K. : Event-Driven Execution Monitor for Ada Tasking Programs, IEEE Proc. of COMPSAC 87, pp. 381-388, 1987.

[Cheng-88] Cheng, J. : Temporal Entailment Logic – Toward a Uniform Logical Basis for Deductive Reasoning Based Concurrent Program Debugging, Seminar memo of Ushijima Lab., Department of Computer Science and Communication Engineering, Kyushu University, Jun. 1988.

[Hughes-72] Hughes, G. E. and Cresswell, M. J. : An Introduction to Model Logic, Hethuen and Co., Ltd. London, 1972.

[Johnstone-87] Johnstone, P. T. : Notes on Logic and Set Theory, Cambridge University Press, 1987.

[Kleene-67] Kleene, S. C. : Introduction to Mathematical logic, John Wiley and Sons, 1967.

[Lin-85a] Lin, B. : Entailment Logic – The Combination of Traditional Logic and Modern Logic, Guizhou, 1985 (in Chinese).

[Lin-85b] Lin, B. : Entailment System – Propositional Calculus System Cm and Notional Calculus System Cn, (an appendix in [Lin-85a], in English).

[Lloyd-87] Lloyd, J. W. : Foundations of Logic Programming, Second Extended Edition, Springer-Verlag, 1987.

[McDermott-80] McDermott, D. and Doyle, J. : Non-Monotonic Logic I , Artificial Intelligence, Vol. 13, No. 1, pp. 41-72, 1980.

[McDermott-82] McDermott, D. : Nonmonotonic Logic II : Nonmonotonic Model Theories, Journal of the Association for Computing Machinery, Vol. 29, No. 1, pp. 33-57, 1982.

[Morgan-76] Morgan, C. G. : Methods for Automated Theorem Proving in Nonclassical Logics, IEEE Transactions on Computers, Vol. C-25, No. 8, pp. 852-862, 1976.

[Nute-80] Nute, D. : Topics in Conditional Logic, D. Reidel, 1980.

[Robinson-79] Robinson, J. A. : Logic : Form and Function – The Mechanization of Deductive Reasoning, Edinburgh University Press, 1979.

[Routley-84] Routley, R. , Meyer, R. K. , Plumwood, V. , and Brady, R. : Relevant Logics and Their Rivals, Part 1 : The Basic Philosophical and Semantical Theory, Ridgeview, Atascadero, California, 1984.

[Rutherford-65] Rutherford, D. E. : Introduction to Lattice Theory, Olliver & boyd, 1965.

[Szasz-63] Szasz, G. : Introduction to Lattice Theory, Academic Press, 1963.

[Thistlewaite-88] Thistlewaite, P. B. , McRobbie, M. A. , and Meyer, R. K. : Automated Theorem Proving in Non-Classical Logics, John Wiley & Sons, 1988.