An Introduction to Classical Propositional Calculus

Jingde Cheng

Saitama University

An Introduction to Classical Propositional Calculus

- Formal (Object) Language of Classical Propositional Calculus (**CPC**)
- Principles of Structural Induction and Structural Recursion
- Model Theory for **CPC**
- Semantic (Model-theoretical or Logical) Consequence Relation
- Normal Forms and Uniform Notation of Formulas
- Proof Theory for **CPC**
- Syntactic (Proof-theoretical or Deductive) Consequence Relation
- Hilbert Style Formal Systems for **CPC**
- Gentzen's Natural Deduction System for **CPC**
- Gentzen's Sequent Calculus System for **CPC**
- Semantic Tableau System for **CPC**
- Resolution System for **CPC**
- Forward Deduction and Backward Deduction

Formal (Object) Language of CPC

- ***Alphabet*** (Symbols)
  - $\{\neg, \rightarrow, \wedge, \vee, \leftrightarrow, \top, \bot, p_1, p_2, ..., p_n, ..., (,)\}$
  - ***Connectives***: $\neg$ (negation), $\rightarrow$ (material implication), $\wedge$ (conjunction), $\vee$ (disjunction), $\leftrightarrow$ (equivalence) .
  - ***Logical constants***: $\top$ and $\bot$.
  - ***Propositional variables*** (letters): $\mathbf{V} =_{df} \{p_1, p_2, ..., p_n, ...\}$.
  - Punctuation: left and right parentheses '(' and ')'.
- Note
  - $A \wedge B =_{df} \neg(A \rightarrow (\neg B))$
  - $A \vee B =_{df} (\neg A) \rightarrow B$
  - $A \rightarrow B =_{df} \neg(A \wedge (\neg B))$ or $(\neg A) \vee B$
  - $A \leftrightarrow B =_{df} (A \rightarrow B) \wedge (B \rightarrow A)$

Formal (Object) Language of CPC

- ***Formulas*** (***Well-formed formulas***)
  - (1) every propositional variable (letter), $\top$, or $\bot$ is a formula (called an ***atomic formula***);
    (2) if A and B are formulas, then so are $(\neg A)$, $(A \rightarrow B)$, $(A \wedge B)$, $(A \vee B)$, $(A \leftrightarrow B)$;
    (3) Nothing else are formulas.
  - $\mathbf{WFF}_{\text{CPC}}$: the set of all formulas of **CPC** (**WFF** for short).
- ***Subformulas***
  - ***Immediate subformulas*** are defined as follows:
    (1) an atomic formula has no immediate subformula;
    (2) the only immediate subformula of $(\neg A)$ is A;
    (3) for a binary connective *, the immediate subformulas of $(A*B)$ are A and B.
  - For any $A \in \mathbf{WFF}$, The set of ***subformulas*** of A is the smallest set S that contains A and contains, with each member, the immediate subformulas of that member. A is called an ***improper subformula*** of itself.

An Introduction to Classical Propositional Calculus

- Formal (Object) Language of Classical Propositional Calculus (CPC)
- Principles of Structural Induction and Structural Recursion
- Model Theory for **CPC**
- Semantic (Model-theoretical or Logical) Consequence Relation
- Normal Forms and Uniform Notation of Formulas
- Proof Theory for **CPC**
- Syntactic (Proof-theoretical or Deductive) Consequence Relation
- Hilbert Style Formal Systems for **CPC**
- Gentzen's Natural Deduction System for **CPC**
- Gentzen's Sequent Calculus System for **CPC**
- Semantic Tableau System for **CPC**
- Resolution System for **CPC**
- Forward Deduction and Backward Deduction

Principles of Structural Induction and Structural Recursion

- ***Principle of Structural Induction***

Every formula in **WFF** has a property, Q, provided:
Basis step: Every atomic formula has property Q;
Induction steps: For any $A \in \mathbf{WFF}$, if A has property Q so does $(\neg A)$; For any $A, B \in \mathbf{WFF}$, if A and B have property Q so does $(A*B)$, where $*$ is a binary connective.

- ***Principle of Structural Recursion***

There is one and only one function f defined on **WFF** such that:
Basis step: The value of f is specified explicitly on atomic formulas;
Recursion steps: For any $A \in \mathbf{WFF}$, the value of f on $(\neg A)$ is specified in terms of the value of f on A; For any $A, B \in \mathbf{WFF}$, the value of f on $(A*B)$ is specified in terms of the values of f on A and on B, where $*$ is a binary connective.

An Introduction to Classical Propositional Calculus

- ♣ Formal (Object) Language of Classical Propositional Calculus (**CPC**)
- ♣ Principles of Structural Induction and Structural Recursion
- ♣ Model Theory for **CPC**
- ♣ Semantic (Model-theoretical or Logical) Consequence Relation
- ♣ Normal Forms and Uniform Notation of Formulas
- ♣ Proof Theory for **CPC**
- ♣ Syntactic (Proof-theoretical or Deductive) Consequence Relation
- ♣ Hilbert Style Formal Systems for **CPC**
- ♣ Gentzen's Natural Deduction System for **CPC**
- ♣ Gentzen's Sequent Calculus System for **CPC**
- ♣ Semantic Tableau System for **CPC**
- ♣ Resolution System for **CPC**
- ♣ Forward Deduction and Backward Deduction

Model Theory for CPC: Model for CPC

♣ Model for **CPC**

A ***model*** for **CPC** is an ordered pair (v_a, v_f) such that
v_a, called a ***truth assignment***, is a function v_a : $\mathbf{V} \rightarrow \{\boldsymbol{t}, \boldsymbol{f}\}$, and
v_f, called a ***truth valuation***, is a function v_f : $\mathbf{WFF} \rightarrow \{\boldsymbol{t}, \boldsymbol{f}\}$ defined as:

(1) for $\top$ and $\bot$, $v_f(\top) = \boldsymbol{t}$ and $v_f(\bot) = \boldsymbol{f}$;

(2) $v_f(A) = v_a(A)$ if $A \in \mathbf{V}$;

(3) $v_f(\neg A) = \boldsymbol{f}$ if $v_f(A) = \boldsymbol{t}$, and $v_f(\neg A) = \boldsymbol{t}$ if $v_f(A) = \boldsymbol{f}$;

(4) $v_f(A \rightarrow B) = \boldsymbol{f}$ if $v_f(A) = \boldsymbol{t}$ and $v_f(B) = \boldsymbol{f}$, and $v_f(A \rightarrow B) = \boldsymbol{t}$ otherwise;

(5) $v_f(A \wedge B) = \boldsymbol{t}$ if both $v_f(A) = \boldsymbol{t}$ and $v_f(B) = \boldsymbol{t}$, and $v_f(A \wedge B) = \boldsymbol{f}$ otherwise;

(6) $v_f(A \vee B) = \boldsymbol{t}$ if $v_f(A) = \boldsymbol{t}$, $v_f(B) = \boldsymbol{t}$ or both, and $v_f(A \vee B) = \boldsymbol{f}$ otherwise;

(7) $v_f(A \leftrightarrow B) = \boldsymbol{t}$ if $v_f(A) = v_f(B)$, and $v_f(A \leftrightarrow B) = \boldsymbol{f}$ otherwise.

Model Theory for CPC: Truth-value, Satisfiability, Validity of a Formula

♣ ***Truth-value*** of a formula in a model

- For any model $M = (v_a, v_f)$ and any $A \in \mathbf{WFF}$, $v_f(A)$ is called the ***truth-value*** of A in M.
- ***Replacement theorem***: Let F, A, $B \in \mathbf{WFF}$, and $F(p)$ mean that p ($p \in \mathbf{V}$) appears in F. If $v_f(A) = v_f(B)$, then $v_f(F(A)) = v_f(F(B))$, where $F(A)$ is the result of substituting A uniformly for p in F, i.e., A replaces every occurrence of p in F.

♣ ***Satisfiability*** of a formula

- For any model $M = (v_a, v_f)$ and any $A \in \mathbf{WFF}$, M ***satisfies*** A or A is ***true*** in M, written as $\models_M A$, iff $v_f(A) = \boldsymbol{t}$; M ***does not satisfy*** A or A is ***false*** in M, written as $\not\models_M A$, iff $v_f(A) = \boldsymbol{f}$.
- For any $A \in \mathbf{WFF}$, A is ***satisfiable*** iff there is some model M such that $\models_M$ A; A is ***unsatisfiable*** iff $\not\models_M A$ for any model M (Ex.: $(A \wedge \neg A)$).

♣ ***Validity*** of a formula

- For any $A \in \mathbf{WFF}$, A is ***valid*** iff $\models_M A$ for any model M (Ex.: $(A \vee \neg A)$).
- Theorem: The validity problem for **CPC**, i.e., whether a formula of **CPC** is a valid or not, is decidable.

Model Theory for CPC: Tautologies, Contradictions, and Contingencies

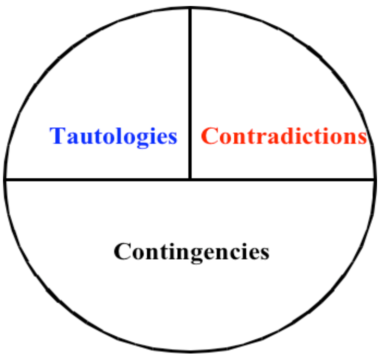
♣ Tautologies, contradictions, and contingencies

- A formula $A \in \mathbf{WFF}$ is a ***tautology*** of **CPC**, written as $\models_{\mathbf{CPC}} A$, iff $\models_M A$ for any model M of **CPC**;
  a formula $A \in \mathbf{WFF}$ is a ***contradiction*** of **CPC**, written as $\not\models_{\mathbf{CPC}} A$, iff $\not\models_M A$ for any model M of **CPC**;
  a formula is a ***contingency*** iff it is neither a tautology nor a contradiction.
- Note: A formula must be any one of tautology, contradiction, and contingency.
- The set of all tautologies of **CPC** is denoted by **Th**(**CPC**).

♣ Relationship between tautologies and contradictions

- Theorem: For any $A \in \mathbf{WFF}$, A is a tautology iff $(\neg A)$ is a contradiction, and A is a contradiction iff $(\neg A)$ is a tautology.
- ***Replacement theorem***: Let F, A, $B \in \mathbf{WFF}$, and F (p) mean that p ($p \in \mathbf{V}$) appears in F. If $(A \leftrightarrow B)$ is a tautology, then so is $(F\ (A) \leftrightarrow F\ (B))$, where F (A) is the result of substituting A uniformly for p in F, i.e., A replaces every occurrence of p in F.

Model Theory for CPC: Tautologies, Contradictions, and Contingencies



Model Theory for CPC: Models of Formulas

♣ Models of formulas

- For any $\Gamma \subseteq \mathbf{WFF}$, model M is called a ***model*** of Γ iff $\models_M A$ for any $A \in \Gamma$.
- The set of all models of Γ is denoted by $\boldsymbol{M}(\Gamma)$.
- $\boldsymbol{M}(\Delta) \subseteq \boldsymbol{M}(\Gamma)$, if $\Gamma \subseteq \Delta$.

♣ ***Consistence*** of formulas

- For any $\Gamma \subseteq \mathbf{WFF}$, Γ is ***semantically (model-theoretically or logically) consistent*** iff it has at least one model; Γ is ***semantically (model-theoretically or logically) inconsistent*** iff it has no model.
- Ex.: $\{A, \neg A, \ldots\}$ is semantically (model-theoretically or logically) inconsistent.

An Introduction to Classical Propositional Calculus

Semantic (Model-theoretical) Logical Consequence Relation

♣ Semantic (model-theoretical or logical) consequence relation

- For any $\Gamma \subseteq$ **WFF** and any $A \in$ **WFF**,
  Γ ***semantically (model-theoretically or logically) entails*** A, or
  A ***semantically (model-theoretically or logically) follows from*** Γ, or
  A is a ***semantic (model-theoretical or logical) consequence*** of Γ,
  written as $\Gamma \models_{\mathbf{CPC}} A$, iff $\models_M A$ for any model M of Γ.

♣ All semantic (model-theoretical or logical) consequences of premises

- The set of all semantic (model-theoretical or logical) consequences of Γ is denoted by $\boldsymbol{C}_{sem}(\Gamma)$.
- $\models_{\mathbf{CPC}} A =_{\mathrm{df}} \phi \models_{\mathbf{CPC}} A$ and it means that $\models_M A$ for any model M of **CPC**, i.e., A is a tautology.

♣ Note

- The semantic (model-theoretical or logical) consequence relation of **CPC** is a semantic (model-theoretical) formalization of the notion that one proposition follows from another or others.

Semantic (Model-theoretical or Logical) Equivalence Relation

♣ Semantic (model-theoretical or logical) equivalence relation

- For any A, $B \in$ **WFF**, A is ***semantically (model-theoretically or logically) equivalent*** to B in **CPC** iff both $\{A\} \models_{\mathbf{CPC}} B$ and $\{B\} \models_{\mathbf{CPC}} A$.
- Theorem: A is semantically (model-theoretically or logically) equivalent to B iff $(A \leftrightarrow B)$ is a tautology.

Properties of the Semantic (Model-theoretical or Logical) Consequence Relation

- If $\models_{\mathbf{CPC}} A$, then $\Gamma \models_{\mathbf{CPC}} A$.
- $\{A\} \models_{\mathbf{CPC}} A$.
- If $A \in \Gamma$, then $\Gamma \models_{\mathbf{CPC}} A$.
- $\Gamma \subseteq \boldsymbol{C}_{sem}(\Gamma)$.
- $\boldsymbol{C}_{sem}(\Gamma) = \boldsymbol{C}_{sem}(\boldsymbol{C}_{sem}(\Gamma))$.
- If $\Gamma \models_{\mathbf{CPC}} A$ and $\Gamma \subseteq \Delta$, then $\Delta \models_{\mathbf{CPC}} A$.
- $\boldsymbol{C}_{sem}(\Gamma) \subseteq \boldsymbol{C}_{sem}(\Delta)$, if $\Gamma \subseteq \Delta$.
- ***Transitivity*** (***the Cut rule***):
  If $\Gamma \models_{\mathbf{CPC}} A$ and $\Delta \cup \{A\} \models_{\mathbf{CPC}} B$, then $\Gamma \cup \Delta \models_{\mathbf{CPC}} B$.
- If $\Gamma \cup \{A_1, ..., A_n\} \models_{\mathbf{CPC}} B$ and $\Gamma \models_{\mathbf{CPC}} A_i$ for $i = 1, ..., n$, then $\Gamma \models_{\mathbf{CPC}} B$.
- ***Substitution***: If $\models_{\mathbf{CPC}} F(p)$, then $\models_{\mathbf{CPC}} F(A)$, where $F(p)$ means that p ($p \in \mathbf{V}$) appears in F ($F \in$ **WFF**) and $F(A)$ is the result of substituting A uniformly for p in F, i.e., A replaces every occurrence of p in F.
- ***Compactness***: $\Gamma \models_{\mathbf{CPC}} A$ iff there is a finite set $\Delta \subseteq \Gamma$ such that $\Delta \models_{\mathbf{CPC}} A$; Γ has a model iff every finite subset of Γ has a model.

Semantic Deduction Theorems

♣ Semantic deduction theorems

- ***Semantic (model-theoretical) deduction theorem for CPC***: For any $A, B \in$ **WFF** and any $\Gamma \subseteq$ **WFF**,
  $$\Gamma \cup \{A\} \models_{\mathbf{CPC}} B \text{ iff } \Gamma \models_{\mathbf{CPC}} (A \rightarrow B).$$
- ***Semantic (model-theoretical) deduction theorem for CPC for finite consequences***: For any $A_1, ..., A_{n-1}, A_n, B \in$ **WFF** and any $\Gamma \subseteq$ **WFF**,
  $$\Gamma \cup \{A_1, ..., A_{n-1}, A_n\} \models_{\mathbf{CPC}} B \text{ iff } \Gamma \models_{\mathbf{CPC}} (A_1 \rightarrow (...(A_{n-1} \rightarrow (A_n \rightarrow B))...));$$
  $$\Gamma \cup \{A_1, ..., A_{n-1}, A_n\} \models_{\mathbf{CPC}} B \text{ iff } \Gamma \models_{\mathbf{CPC}} ((A_1 \wedge (...(A_{n-1} \wedge A_n)...)) \rightarrow B).$$

♣ Notes

- As a special case of the above deduction theorems, $\{A\} \models_{\mathbf{CPC}} B$ iff $\models_{\mathbf{CPC}} (A \rightarrow B)$, i.e., A semantically (model-theoretically or logically) entails B iff $(A \rightarrow B)$ is a tautology.
- In the framework of **CPC**, the semantic (model-theoretical or logical) consequence relation, which is a representation of the notion of entailment in the sense of meta-logic, is "equivalent" to the notion of material implication.

An Introduction to Classical Propositional Calculus

Normal Forms of Formulas: Literal and Clause

♣ Generalized disjunction and conjunction

Let $X_1, X_2, ..., X_n$ be a list of formulas.

- ***Generalized disjunction***: $[X_1, X_2, ..., X_n] =_{df} X_1 \vee X_2 \vee ... \vee X_n$.
- ***Generalized conjunction***: $<X_1, X_2, ..., X_n> =_{df} X_1 \wedge X_2 \wedge ... \wedge X_n$.

♣ Literal

- A formula is called a ***literal***, if it is a propositional variable or the negation of a propositional variable, or a constant, $\top$ or $\bot$.

♣ Clause

- A ***clause*** is a generalized disjunction $[X_1, X_2, ..., X_n]$ in which each member is a literal.
- A ***dual clause*** is a generalized conjunction $<X_1, X_2, ..., X_n>$ in which each member is a literal.

Normal Forms of Formulas: CNF and DNF

♣ Conjunctive normal form

- A formula is called a ***formula in conjunctive normal form*** or a ***formula in clause form*** or a ***clause set***, if it is a generalized conjunction $<C_1, C_2, ..., C_n>$ in which each member is a clause.
- A ***conjunctive normal form*** (***CNF***) for a formula A is a formula B in conjunctive normal form such that B contains exactly the same propositional variables in A and is semantically equivalent to A.

♣ Disjunctive normal form

- A formula is called a ***formula in disjunctive normal form*** or a ***formula in dual clause form*** or a ***dual clause set***, it is a generalized disjunction $[D_1, D_2, ..., D_n]$ in which each member is a dual clause.
- A ***disjunctive normal form*** (***DNF***) for a formula A is a formula B in disjunctive normal form such that B contains exactly the same propositional variables in A and is semantically equivalent to A.

♣ The ***normal form theorem*** for **CPC**

- Theorem: There are algorithms for converting an ordinary formula into its conjunctive normal form and its disjunctive normal form.

Uniform Notation of Formulas

♣ Uniform notation of formulas [R. M. Smullyan, 1968]

- Classify all formulas of the forms $(A*B)$ and $(\neg(A*B))$, where $*$ is a binary connective, into two categories, i.e., ***α-formulas*** which act conjunctively, and ***β-formulas***, which act disjunctively.
- For each α formula, we define two components, which we denote α_1 and α_2. For each β-formula, we define two components, which we denote β_1 and β_2.

♣ β formulas of “$\leftrightarrow$“

- If a β-formula is of form $(A\leftrightarrow B)$, then β_1 denotes $(A\wedge B)$ and β_2 denotes $((\neg A)\wedge(\neg B))$
- If a β-formula is of form $(\neg(A\leftrightarrow B))$, then β_1 denotes $((\neg A)\wedge B)$ and β_2 denotes $(A\wedge(\neg B))$.

♣ Theorems

- For any model, an α-formula is true iff both α_1 and α_2 true; a β-formula is true iff β_1 or β_2 is true.
- For any α and β, $(\alpha\leftrightarrow(\alpha_1\wedge\alpha_2))$ and $(\beta\leftrightarrow(\beta_1\vee\beta_2))$ are tautologies.

α-Formulas and β-Formulas and Their Components

Conjunctive			Disjunctive		
α	α_1	α_2	β	β_1	β_2
$X\wedge Y$	X	Y	$\neg(X\wedge Y)$	$\neg X$	$\neg Y$
$\neg(X\vee Y)$	$\neg X$	$\neg Y$	$X\vee Y$	X	Y
$\neg(X\rightarrow Y)$	X	$\neg Y$	$X\rightarrow Y$	$\neg X$	Y
$\neg(X\leftarrow Y)$	$\neg X$	Y	$X\leftarrow Y$	X	$\neg Y$
$\neg(X\neg\wedge Y)$	X	Y	$X\neg\wedge Y$	$\neg X$	$\neg Y$
$X\neg\vee Y$	$\neg X$	$\neg Y$	$\neg(X\neg\vee Y)$	X	Y
$X\neg\rightarrow Y$	X	$\neg Y$	$\neg(X\neg\rightarrow Y)$	$\neg X$	Y
$X\neg\leftarrow Y$	$\neg X$	Y	$\neg(X\neg\leftarrow Y)$	X	$\neg Y$

An Introduction to Classical Propositional Calculus

♣ Formal (Object) Language of Classical Propositional Calculus (**CPC**)
♣ Principles of Structural Induction and Structural Recursion
♣ Model Theory for **CPC**
♣ Semantic (Model-theoretical or Logical) Consequence Relation
♣ Normal Forms and Uniform Notation of Formulas
♣ Proof Theory for **CPC**
♣ Syntactic (Proof-theoretical or Deductive) Consequence Relation
♣ Hilbert Style Formal Systems for **CPC**
♣ Gentzen's Natural Deduction System for **CPC**
♣ Gentzen's Sequent Calculus System for **CPC**
♣ Semantic Tableau System for **CPC**
♣ Resolution System for **CPC**
♣ Forward Deduction and Backward Deduction

Proof Theory: Formal System, Proof and Theorem

♣ Formal system

- A ***formal system*** has the following components:
  (1) alphabet: a non-empty set of symbols,
  (2) grammar: a finite set of rules for forming formulas,
  (3) axioms: a set of formulas as start points for deduction, and
  (4) deduction (inference) rules: a finite set of rules for generating a new formula (the consequence) from some old (the premises and/or hypotheses).

♣ Proof and theorem

- A ***proof*** of f_n in a formal system is a finite sequence of formulas $f_1, ..., f_n$ such that, for all i ($i \leq n$), (1) f_i is an axiom, or (2) there are some members $f_{j1}, ..., f_{jm}$ ($j1, ..., jm<i$) of the sequence, which have f_i as the result of applying one of the deduction rules to $f_{j1}, ..., f_{jn}$.
- If $f_1, ..., f_n$ is a proof in a formal system, then f_n is called a ***theorem*** of the formal system and said to be ***provable*** in the formal system.
- The set of all theorems of a formal system **FS** is denoted by **Th**(**FS**).

Proof Theory: Deduction

♣ Deduction

- Let P be a set of formulas in a formal system. A ***deduction*** (***proof***) ***from*** P in the formal system is a finite sequence of formulas $f_1, ..., f_n$ such that, for all i ($i \leq n$), (1) f_i is an axiom, or (2) $f_i \in P$, or (3) there are some members $f_{j1}, ..., f_{jm}$ ($j1, ..., jm < i$) of the sequence, which have f_i as the result of applying one of the deduction rules to $f_{j1}, ..., f_{jm}$.
- If $f_1, ..., f_n$ is a deduction (proof) from P in a formal system, then P is called the ***premises*** of the deduction and f_n is called the ***consequence*** and said to be ***deducible from*** P in the formal system.
- A theorem of a formal system is deducible from the empty premises in the formal system.

An Introduction to Classical Propositional Calculus

♣ Formal (Object) Language of Classical Propositional Calculus (**CPC**)
♣ Principles of Structural Induction and Structural Recursion
♣ Model Theory for **CPC**
♣ Semantic (Model-theoretical or Logical) Consequence Relation
♣ Normal Forms and Uniform Notation of Formulas
♣ Proof Theory for **CPC**
♣ Syntactic (Proof-theoretical or Deductive) Consequence Relation
♣ Hilbert Style Formal Systems for **CPC**
♣ Gentzen's Natural Deduction System for **CPC**
♣ Gentzen's Sequent Calculus System for **CPC**
♣ Semantic Tableau System for **CPC**
♣ Resolution System for **CPC**
♣ Forward Deduction and Backward Deduction

Syntactic (Proof-theoretical or Deductive) Consequence Relation

♣ Syntactic (proof-theoretical or deductive) consequence relation

- In a formal system **FS**, for any set Γ of formulas and any formula A, Γ ***syntactically (proof-theoretically or deductively) entails*** A, or A ***syntactically (proof-theoretically or deductively) follows from*** Γ, or A is a ***syntactic (proof-theoretical or deductive) consequence*** of Γ, written as $\Gamma \vdash_{\mathbf{FS}} A$ ("$\vdash$" is read as "turnstile"), iff A is deducible from Γ in **FS**.
- The set of all syntactic (proof-theoretical or deductive) consequences of Γ is denoted by $C_{syn}(\Gamma)$.
- $\vdash_{\mathbf{FS}} A =_{\mathrm{df}} \phi \vdash_{\mathbf{FS}} A$ and it means that A is a theorem of **FS**.
- The syntactic (proof-theoretical or deductive) consequence relation is a syntactic (proof-theoretical) formalization of the notion that one proposition follows from another or others.

♣ Syntactic (proof-theoretical or deductive) equivalence relation

- For any two formulas A and B, A is ***syntactically (proof-theoretically or deductively) equivalent*** to B in **FS** iff both $\{A\} \vdash_{\mathbf{FS}} B$ and $\{B\} \vdash_{\mathbf{FS}} A$.

Closure and Theory

♣ Closure

- In a formal system **FS**, a set Σ of formulas is ***closed under*** a deduction rule of **FS** if whenever the premises of the rule are in Σ, then so is the consequence.
- In a formal system **FS**, the ***closure*** of a set Σ of formulas under a rule is the smallest set Δ of formulas such that $\Delta \supseteq \Sigma$ and Δ is closed under the rule.

♣ Theory

- In a formal system **FS**, for a set Σ of formulas, the set of all syntactic consequences of Σ is called the ***theory of*** Σ, denoted $Th(\Sigma)$, i.e., $Th(\Sigma) =_{\mathrm{df}} \{A \mid \Sigma \vdash_{\mathbf{FS}} A\}$.
- In a formal system **FS**, a set Σ of formulas is called a ***theory*** if it is closed under the relation of syntactic consequence, i.e., $Th(\Sigma) = \Sigma$.

Properties of the Syntactic (Proof-theoretical or Deductive) Consequence Relation

- If $\vdash_{\mathbf{FS}} A$, then $\Gamma \vdash_{\mathbf{FS}} A$.
- $\{A\} \vdash_{\mathbf{FS}} A$.
- If $A \in \Gamma$, then $\Gamma \vdash_{\mathbf{FS}} A$.
- If $\Gamma \vdash_{\mathbf{FS}} A$ and $\Gamma \subseteq \Delta$, then $\Delta \vdash_{\mathbf{FS}} A$.
- ***Transitivity*** (***the Cut rule***): If $\Gamma \vdash_{\mathbf{FS}} A$ and $\Delta \cup \{A\} \vdash_{\mathbf{FS}} B$, then $\Gamma \cup \Delta \vdash_{\mathbf{FS}} B$.
- If $\Gamma \cup \{A_1, ..., A_n\} \vdash_{\mathbf{FS}} B$ and $\Gamma \vdash_{\mathbf{FS}} A_i$ for $i = 1, ..., n$, then $\Gamma \vdash_{\mathbf{FS}} B$.
- ***Compactness***: $\Gamma \vdash_{\mathbf{FS}} A$ iff there is a finite set $\Delta \subseteq \Gamma$ such that $\Delta \vdash_{\mathbf{FS}} A$.

Properties of the Syntactic (Proof-theoretical or Deductive) Consequence Relation

♣ Post-consistence and Post-completeness

Let **FS** be a formal system and Γ be a set of formulas.

- Γ is ***Post-consistent*** (***syntactically consistent***) in **FS** iff there is some formula A such that $\Gamma \vdash_{\mathbf{FS}} A$ does not hold.
- Γ is ***Post-complete*** (***syntactically complete***) in **FS** iff for every formula B not in Γ and for every C, $\Gamma \cup \{B\} \vdash_{\mathbf{FS}} C$.

♣ Theorems

- If Γ is Post-consistent in **FS** and $\Delta \subseteq \Gamma$ then Δ is Post-consistent in **FS**.
- Γ is Post-consistent in **FS** iff every finite subset of Γ is Post-consistent in **FS**.
- If Γ is both Post-complete and Post-consistent in **FS**, then Γ is a theory.

Properties of the Syntactic (Proof-theoretical or Deductive) Consequence Relation

♣ Classical-consistence and Classical-completeness

Let **FS** be a formal system and Γ be a set of formulas.

- Γ is ***classically consistent*** in **FS** iff for every formula A, not both $\Gamma \vdash_{\mathbf{FS}} A$ and $\Gamma \vdash_{\mathbf{FS}} (\neg A)$.
- Γ is ***classically complete*** **FS** iff for every formula A, at least one of A and $(\neg A)$ in Γ.

♣ Theorems

- Γ is classically consistent in **FS** iff Γ is Post-consistent in **FS**.
- If Γ is a theory, then Γ is classically complete in **FS** iff Γ is Post-complete in **FS**.

An Introduction to Classical Propositional Calculus

Hilbert Style Formal Systems

♣ Hilbert style formal systems

- The most classical (historical) style of formal systems, which was first given by G. Frege in 1879.
- The mechanism of a Hilbert style formal system works in the forward deduction principle.
- For certain philosophical logics, only Hilbert style formulations are known to exist.
- Hilbert style formal systems are widespread, and should be familiar to everyone who uses formal logic.

♣ ***Forward deduction principle***

- To prove a formula in a formal system, one starts with some formulas as premises which are known facts or assumed hypotheses, derives immediate consequences, immediate consequences of the immediate consequences, and so on by applying inference rules, until the desired formula is reached.

L: A Hilbert Style Formal System for CPC

♣ Alphabet

$\{\neg, \rightarrow, p_1, p_2, ..., p_n, ..., (,)\}$

♣ Formulas (Well-formed formulas) of **L**: $\mathbf{WFF}_{\mathbf{L}}$

(1) $p_1, p_2, ..., p_n, ...$ are (atomic) formulas;
(2) if A and B are formulas, then so are $(\neg A)$ and $(A \rightarrow B)$;
(3) Nothing else are formulas.

♣ Axiom schemata of **L** [Lukasiewicz, 1930]

$(A \rightarrow (B \rightarrow A))$
$((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$
$(((\neg A) \rightarrow (\neg B)) \rightarrow (B \rightarrow A))$

♣ Inference rule of **L**

Modus Ponens for material implication: from A and $(A \rightarrow B)$ to infer B.

Properties of L

♣ ***Syntactic (proof-theoretical) deduction theorems for L***

- For any $A, B \in \mathbf{WFF}_{\mathbf{L}}$ and any $\Gamma \subseteq \mathbf{WFF}_{\mathbf{L}}$, $\Gamma \cup \{A\} \vdash_{\mathbf{L}} B$ iff $\Gamma \vdash_{\mathbf{L}} A \rightarrow B$.
- For any $A_1, ..., A_{n-1}, A_n, B \in \mathbf{WFF}_{\mathbf{L}}$ and any $\Gamma \subseteq \mathbf{WFF}_{\mathbf{L}}$,
  $\Gamma \cup \{A_1, ..., A_{n-1}, A_n\} \vdash_{\mathbf{L}} B$ iff $\Gamma \vdash_{\mathbf{L}} (A_1 \rightarrow (...(A_{n-1} \rightarrow (A_n \rightarrow B))...))$;
  $\Gamma \cup \{A_1, ..., A_{n-1}, A_n\} \vdash_{\mathbf{L}} B$ iff $\Gamma \vdash_{\mathbf{L}} ((A_1 \wedge (...(A_{n-1} \wedge A_n)...)) \rightarrow B)$.

♣ Notes

- $\{A\} \vdash_{\mathbf{L}} B$ iff $\vdash_{\mathbf{L}} A \rightarrow B$, i.e., B syntactically (proof-theoretically or deductively) entails A iff $A \rightarrow B$ is a theorem. This means that in the framework of **CPC**, the syntactic (proof-theoretical or deductive) consequence relation, which is the notion of conditional in the sense of meta-logic, is "equivalent" to the notion of material implication.
- The above deduction theorems are also true for any Hilbert style formal system with at least axiom schemata "$(A \rightarrow (B \rightarrow A))$" and "$((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$" and with Modus Ponens for material implication as the only inference rule.

Properties of L: Soundness and Completeness

♣ ***Soundness theorems for L***

- Theorem (***soundness***): If $\vdash_{\mathbf{L}} A$ then $\models_{\mathbf{CPC}} A$, for any $A \in \mathbf{WFF}_{\mathbf{L}}$, i.e., for any $A \in \mathbf{WFF}_{\mathbf{L}}$, $A \in \mathbf{Th}(\mathbf{CPC})$ if $A \in \mathbf{Th}(\mathbf{L})$, $\mathbf{Th}(\mathbf{L}) \subseteq \mathbf{Th}(\mathbf{CPC})$.
- Theorem (***strong soundness***): If $\Gamma \vdash_{\mathbf{L}} A$ then $\Gamma \models_{\mathbf{CPC}} A$, for any $A \in \mathbf{WFF}$ and any $\Gamma \subseteq \mathbf{WFF}$.

♣ ***Completeness theorems for L***

- Theorem (***completeness***): If $\models_{\mathbf{CPC}} A$ then $\vdash_{\mathbf{L}} A$, for any $A \in \mathbf{WFF}$, i.e., for any $A \in \mathbf{WFF}$, $A \in \mathbf{Th}(\mathbf{L})$ if $A \in \mathbf{Th}(\mathbf{CPC})$, $\mathbf{Th}(\mathbf{CPC}) \subseteq \mathbf{Th}(\mathbf{L})$.
- Theorem (***strong completeness***): If $\Gamma \models_{\mathbf{CPC}} A$ then $\Gamma \vdash_{\mathbf{L}} A$, for any $A \in \mathbf{WFF}$ and any $\Gamma \subseteq \mathbf{WFF}$.

♣ CPC vs. L

- $\mathbf{Th}(\mathbf{CPC}) = \mathbf{Th}(\mathbf{L})$.

Other Hilbert Style Axiomatizations for CPC

♣ Frege's system [G. Frege, 1879]

- $(A \to (B \to A))$
  $((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$
  $((A \to (B \to C)) \to (B \to (A \to C)))$ (is deducible from the above two)
  $((A \to B) \to ((\neg B) \to (\neg A)))$
  $((\neg(\neg A)) \to A)$
  $(A \to (\neg(\neg A)))$ (the last three can be replaced by
  $(((\neg A) \to (\neg B)) \to (B \to A))$
- L [Lukasiewicz, 1930] is a simplification of Frege's system.

Other Hilbert Style Axiomatizations for CPC

♣ Hilbert and Bernays' system [D. Hilbert and P. Bernays, 1934]

- $(A \to (B \to A))$
  $((A \to (A \to B)) \to (A \to B))$
  $((A \to B) \to ((B \to C) \to (A \to C)))$
- $((A \wedge B) \to A)$, $((A \wedge B) \to B)$
  $((A \to B) \to ((A \to C) \to (A \to (B \wedge C))))$
- $(A \to (A \vee B))$, $(B \to (A \vee B))$
  $((A \to C) \to ((B \to C) \to ((A \vee B) \to C)))$
- $((A \leftrightarrow B) \to (A \to B))$, $((A \leftrightarrow B) \to (B \to A))$
  $((A \to B) \to ((B \to A) \to (A \leftrightarrow B)))$
- $((A \to B) \to ((\neg B) \to (\neg A)))$
  $(A \to (\neg(\neg A)))$, $((\neg(\neg A)) \to A)$

An Example of Deduction in L

$\{A, B\} \vdash_L (A \wedge B)$?

A
B
$(A \to (A \to A))$ $\{(A \to (B \to A))\}$
$(A \to A)$
$((A \to A) \to ((A \to B) \to (A \to (A \wedge B))))$
$\{((A \to B) \to ((A \to C) \to (A \to (B \wedge C))))\}$
$((A \to B) \to (A \to (A \wedge B)))$
$(B \to (A \to B))$ $\{(A \to (B \to A))\}$
$(A \to B)$
$(A \to (A \wedge B))$
$(A \wedge B)$

An Example of Theorem Proof in L

$\vdash_L (A \to (B \to (A \wedge B)))$?

By $\{A, B\} \vdash_L (A \wedge B)$ and syntactic (proof-theoretical) deduction theorems for **L**, we can directly have $\vdash_L (A \to (B \to (A \wedge B)))$.

An Introduction to Classical Propositional Calculus

♣ Formal (Object) Language of Classical Propositional Calculus (**CPC**)
♣ Principles of Structural Induction and Structural Recursion
♣ Model Theory for **CPC**
♣ Semantic (Model-theoretical or Logical) Consequence Relation
♣ Normal Forms and Uniform Notation of Formulas
♣ Proof Theory for **CPC**
♣ Syntactic (Proof-theoretical or Deductive) Consequence Relation
♣ Hilbert Style Formal Systems for **CPC**
♣ Gentzen's Natural Deduction System for **CPC**
♣ Gentzen's Sequent Calculus System for **CPC**
♣ Semantic Tableau System for **CPC**
♣ Resolution System for **CPC**
♣ Forward Deduction and Backward Deduction