An Introduction to Classical Predicate Calculus

Jingde Cheng

Saitama University

An Introduction to Classical Predicate Calculus

- ♣ The Limitations of Propositional Logic **CPC**
- ♣ Formal (Object) Language of Classical First Order Predicate Calculus (**CFOPC**)
- ♣ Substitutions
- ♣ Model Theory for **CFOPC**
- ♣ Semantic (Model-theoretical or Logical) Consequence Relation
- ♣ Hilbert Style Formal Systems for **CFOPC**
- ♣ Gentzen's Natural Deduction System for **CFOPC**
- ♣ Gentzen's Sequent Calculus System for **CFOPC**
- ♣ Semantic Tableau Systems for **CFOPC**
- ♣ Resolution Systems for **CFOPC**
- ♣ Second Order Predicate Calculus

The Limitations of Propositional Logic CPC

♣ Example

- All men are mortal
  Socrates is a man
  Therefore:
  Socrates is mortal
- A: All *M* are *P*
  B: *S* is a *M*
  Therefore:
  C: *S* is *P*
- However, $((A \wedge B) \rightarrow C)$ is not a tautology in propositional logic **CPC**.

♣ Example

- A: $\forall x(M(x) \rightarrow P(x))$
  B: $\exists s(M(s))$
  Therefore:
  C: $P(s)$
- $((A \wedge B) \rightarrow C)$ is a tautology in first order predicate logic.

An Introduction to Classical Predicate Calculus

- ♣ The Limitations of Propositional Logic **CPC**
- ♣ Formal (Object) Language of Classical First Order Predicate Calculus (**CFOPC**)
- ♣ Substitutions
- ♣ Model Theory for **CFOPC**
- ♣ Semantic (Model-theoretical or Logical) Consequence Relation
- ♣ Hilbert Style Formal Systems for **CFOPC**
- ♣ Gentzen's Natural Deduction System for **CFOPC**
- ♣ Gentzen's Sequent Calculus System for **CFOPC**
- ♣ Semantic Tableau Systems for **CFOPC**
- ♣ Resolution Systems for **CFOPC**
- ♣ Second Order Predicate Calculus

Formal (Object) Language of CFOPC

♣ ***Alphabet*** (Symbols)

- $\{\neg, \rightarrow, \wedge, \vee, \leftrightarrow, \forall, \exists, \top, \bot, (,),$
  $x_1, x_2, ..., x_n, ...,$
  $c_1, c_2, ..., c_n, ...,$
  $f_1^1, ..., f_1^n, ..., f_2^1, ..., f_2^n, ..., f_k^1, ..., f_k^n, ...,$
  $p_0^1, ..., p_0^n, ..., p_1^1, ..., p_1^n, ..., p_2^1, ..., p_2^n, ..., p_k^1, ..., p_k^n, ... \}$
- ***Connectives***: $\neg$ (negation), $\rightarrow$ (material implication), $\wedge$ (conjunction), $\vee$ (disjunction), $\leftrightarrow$ (equivalence).
- ***Quantifiers***: $\forall$ (for all, the ***universal quantifier***), $\exists$ (there exists, the ***existential quantifier***).
- Logical constants: $\top$ and $\bot$.
- Punctuation: left and right parentheses (and).

Formal (Object) Language of CFOPC

♣ ***Alphabet*** (Symbols)

- ***Individual variables (variable symbols)***
  (**V**): $x_1, x_2, ..., x_n, ...$.
- Note: Individual variables, which rang over the domain of discourse, act as placeholders in much the same way as pronouns act as placeholders in ordinary language.
- ***(Individual) Constants (Names) (constant symbols, name symbols)***
  (**C**): $c_1, c_2, ..., c_n, ...$
- Note: Constants play the role of names for objects (individuals) in the domain of discourse.

Formal (Object) Language of CFOPC

♣ ***Alphabet*** (Symbols)

- ***(Individual) Functions (function symbols)***

  (**F**): $f_1^1, \ldots, f_1^n, \ldots, f_2^1, \ldots, f_2^n, \ldots, f_k^1, \ldots, f_k^n, \ldots$

- Note: Constants can be regarded as 0-ary functions because they are objects (individuals) that have no dependence on any inputs; they simply denote objects (individuals) of the domain of discourse.
- ***(Individual) Predicates (Relations) (predicate symbols, relation symbols)***

  (**P**): $p_0^1, \ldots, p_0^n, \ldots, p_2^1, \ldots, p_2^n, \ldots, p_k^1, \ldots, p_k^n, \ldots$

- Note: 0-ary predicates can be regarded as propositions (sentences) because they are simply statements of facts independent of any individual variables. Unary predicates are simply properties of objects (individuals), binary predicates are relations between pairs of objects (individuals), and in general n-ary predicates express relations among n-tuple of objects (individuals).

Formal (Object) Language of CFOPC

♣ ***Terms***

- (1) Every individual variable (symbol) is a term;
  (2) Every constant (symbol) is a term;
  (3) If f is an n-ary function (symbol) ($n = 1, 2, \ldots$) and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term;
  (4) Nothing else are terms.
- **T**: the set of all terms

♣ ***Closed terms***

- A term is ***closed***, called a ***variable-free term*** or ***ground term***, iff it contains no individual variables.

Formal (Object) Language of CFOPC

♣ ***Formulas*** (Well-formed formulas)

- (1) If p is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms, then $p(t_1, \ldots, t_n)$ is a formula (called an ***atomic formula***);
  (2) If A and B are formulas and x is an individual variable, then so are $(\neg A)$, $(A \rightarrow B)$, $(A \wedge B)$, $(A \vee B)$, $(A \leftrightarrow B)$, $(\forall x A)$, and $(\exists x A)$;
  (3) Nothing else are formulas.
- $\mathbf{WFF}_{\mathrm{CFOPC}}$: the set of all formulas of **CFOPC** (**WFF** for short).

♣ Subformulas

- A ***subformula*** of a formula f is a consecutive sequence of symbols from f which is itself a formula.

♣ Open formulas

- An ***open formula*** is a formula without quantifiers.

An Introduction to Classical Predicate Calculus

Free Occurrence and Bound Occurrence

♣ ***Free occurrences*** of individual variables

(1) The free variable occurrences in an atomic formula are all the variable occurrences in the formula.

(2) The free variable occurrences in $(\neg A)$ are the free variable occurrences in A.

(3) The free variable occurrences in $(A*B)$ are the free variable occurrences in A together with the free variable occurrences in B, where * is a binary connective.

(4) The free variable occurrences in $(\forall x A)$ and $(\exists x A)$ are the free variable occurrences in A, except for occurrences of x.

♣ ***Bound occurrences*** of individual variables

- A variable occurrence is bound iff it is not free.
- A formula with no free variable occurrences is called a ***sentence*** (***closed formula***).

Substitutions

♣ ***Substitution*** of variable

- A variable ***substitution*** is a mapping σ: $\mathbf{V} \rightarrow \mathbf{T}$ from the set of individual variables **V** to the set of terms **T**; it can be extended to all terms and formulas as follows (we denote $\sigma[x]$ by $x\sigma$, and so on) :
  (1) $c\sigma = c$ for any $c \in \mathbf{C}$;
  (2) $x\sigma = x\sigma$ for any $x \in \mathbf{V}$;
  (3) $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$ for any $f \in \mathbf{F}$;
  (4) $p(t_1, \ldots, t_n)\sigma = p(t_1\sigma, \ldots, t_n\sigma)$ for any $p \in \mathbf{P}$;
  (5) $(\neg A)\sigma = (\neg(A\sigma))$ for any $A \in \mathbf{WFF}$;
  (6) $(A*B)\sigma = ((A\sigma)*(B\sigma))$ for any A, $B \in \mathbf{WFF}$, where * is a binary connective;
  (7) $(\forall x A)\sigma = (\forall x(A\sigma_x))$ and $(\exists x A)\sigma = (\exists x(A\sigma_x))$ for any $A \in \mathbf{WFF}$, where by σ_x we mean the substitution that is like σ except that it does not change x, i.e., $y\sigma_x = y\sigma$ if $y \neq x$ and $y\sigma_x = x$ if $y = x$.
- Note: The result of applying a substitution to a term always producers another term.

Substitutions

♣ ***Composition*** of substitutions

- Let σ and τ be substitutions. By the ***composition*** of σ and τ, we mean that substitution, which we denote by $\sigma \bullet \tau$, such that for each variable $x \in$ **V**, $x(\sigma \bullet \tau) = (x\sigma)\tau$.
- Theorem: For any term $t \in$ **T** and any substitutions σ and τ, $t(\sigma \bullet \tau) = (t\sigma)\tau$.
- Theorem: Composition of substitutions is associative, i.e., for any substitutions σ_1, σ_2, and σ_3, $(\sigma_1 \bullet \sigma_2) \bullet \sigma_3 = \sigma_1 \bullet (\sigma_2 \bullet \sigma_3)$.

♣ ***Support*** of substitution

- The ***support*** of a substitution σ is the set of variables x for which $x\sigma \neq x$. A substitution has ***finite support*** if its support set is finite.
- Theorem: The composition of two substitutions having finite support is a substitution having finite support.

♣ Notation of substitution

- Suppose σ is a substitution having finite support; say $[x_1, x_2, ..., x_n]$ is the support, and for each $i = 1, ..., n$, $x_i\sigma = t_i$. Our notation for σ is: $[x_1/t_1, x_2/t_2, ..., x_n/t_n]$. In particular, our notation for the identity substitution is [ ].

Substitutions

♣ Example

- Let $\sigma = [x/y]$ and $\tau = [y/c]$. Then $\sigma \bullet \tau = [x/y, y/c]$.
  If $A = (\forall y R(x, y))$, then $A\sigma = (\forall y R(y, y))$, so $(A\sigma)\tau = (\forall y R(y, y))$.
  But $A(\sigma \bullet \tau) = (\forall y R(c, y))$, which is different.
- The fact about substitution in terms that for any term t, $(t\sigma)\tau = t(\sigma \bullet \tau)$ does not carry over to formulas.
- What is needed is some restriction that will ensure composition of substitutions behaves well.

Substitutions

♣ ***Free substitution***

- A substitution being ***free for a formula*** is characterized as follows:

(1) If $A \in$ **WFF** is an atomic formula, then σ is free for A.

(2) For any $A \in$ **WFF**, σ is free for $\neg A$, if σ is free for A.

(3) For any $A, B \in$ **WFF**, σ is free for $(A*B)$, if σ is free for A and σ is free for B, where * is a binary connective.

(4) For any $A \in$ **WFF**, σ is free for $(\forall xA)$ and $(\exists xA)$ provided: σ_x is free for A, and if y is a free variable of A other than x, $y\sigma$ does not contain x.

- Theorem: Suppose the substitution σ is free for the formula A, and the substitution τ is free for $A\sigma$. Then $(A\sigma)\tau = A(\sigma \bullet \tau)$.

An Introduction to Classical Predicate Calculus

Model Theory for CFOPC: Models (Structures)

♣ ***Models*** (***Structures***) for first order languages

- Let L(**C**, **F**, **P**) is a first order language determined by **C**, **F**, **P**. A ***model*** (***structure***) for L(**C**, **F**, **P**) is an ordered pair $M = (D, I)$ where D is a non-empty set, called the ***domain*** or ***universe*** of M and I is a mapping, called an ***interpretation*** such that:
  for every constant symbol $c \in$ **C**, $c^I \in D$;
  for every n-ary function symbol $f \in$ **F**,
  f^I is an n-ary function on D, $f^I : D^n \rightarrow D$;
  for every n-ary predicate symbol $p \in$ **P**,
  p^I is an n-ary relation on D, $p^I \subseteq D^n$.
- An ***assignment*** ***Ass*** in a model $M = (D, I)$ is a mapping from the set of individual variables **V** to the domain D. The image of the individual variable x under the assignment Ass is denoted by x^{Ass}.

♣ Note

- A model for the first order language L(**C**, **F**, **P**) together with an assignment in the model give an interpretation for the language.

Model Theory for CFOPC: Interpretations for Terms

♣ ***Interpretations*** for terms

- Let $M = (D, I)$ be a model of the first order language L(**C**, **F**, **P**), and let $\boldsymbol{A}$ be an assignment in the model. For every term $t \in$ **T**, its interpretation (a value in D) is defined as follows:

(1) $c^{IA} = c^I$ for every $c \in$ **C**.

(2) $x^{IA} = x^A$ for every $x \in$ **V**.

(3) $[f(t_1, ..., t_n)]^{IA} = f^I(t_1^{IA}, ..., t_n^{IA})$ for every $f \in$ **F**.

- Note: The value of a closed term does not depend on the assignment $\boldsymbol{A}$.

♣ ***Variant*** of assignment

- Let $M = (D, I)$ be a model of the first order language L(**C**, **F**, **P**), and let $x \in$ **V** be an individual variable. The assignment $\boldsymbol{B}$ in the model M is an ***x-variant*** of the assignment $\boldsymbol{A}$, if $\boldsymbol{A}$ and $\boldsymbol{B}$ assign the same values to every individual variable in **V** except possibly x.

Model Theory for CFOPC: Truth-Value of Formula

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

- Let $M = (D, I)$ be a model of L(**C**, **F**, **P**), and let ***A*** be an assignment in the model. For any $R \in$ **WFF**, its ***truth-value*** $v_f^{IA}(R)$ under ***A*** in M is defined by a ***truth valuation*** function v_f^{IA} : **WFF**→{***t***, ***f***} as follows:

(1) for every atomic formula $p\ (t_1, ..., t_n) \in$ **WFF**, $v_f^{IA}(p\ (t_1, ..., t_n)) =$ ***t***

if $(t_1^{IA}, ..., t_n^{IA}) \in p^{\,I}$, and $v_f^{IA}(p\ (t_1, ..., t_n)) =$ ***f*** otherwise;

(2) for any $(\neg R)$, $(R*S) \in$ **WFF**, where * is a binary connective,

the same as the definition of v_f of **CPC**;

(3) for any $(\forall xR)$, $v_f^{IA}((\forall xR)) =$ ***t*** if $v_f^{IB}(R) =$ ***t*** for every assignment

B in M that is an x-variant of ***A***, and $v_f^{IA}((\forall xR)) =$ ***f*** otherwise;

(4) for any $(\exists xR)$, $v_f^{IA}((\exists xR)) =$ ***t*** if $v_f^{IB}(R) =$ ***t*** for some assignment

B in M that is an x-variant of ***A***, and $v_f^{IA}((\forall xR)) =$ ***f*** otherwise.

- Note: The truth-value of a sentence (closed formula) does not depend on the assignment ***A***.

Model Theory for CFOPC: Satisfiability of Formula

♣ ***Satisfiability*** of a formula in a model

For any model $M = (D, I)$ of the first order language L(**C**, **F**, **P**) and any $R \in$ **WFF**,

- R is ***satisfiable*** in M or R ***may be true*** in M iff there is some assignment ***A*** (called a ***satisfying assignment***) such that under A, $v_f^{IA}(R) =$ ***t***;
- M ***satisfies*** R or R is ***true*** in M, written as $\models_M R$, iff $v_f^{IA}(R) =$ ***t*** for any assignment ***A***;
- M ***does not satisfy*** R or R ***may be false*** in M iff there is some assignment ***A*** such that under ***A***, $v_f^{IA}(R) =$ ***f***;
- R is ***unsatisfiable*** in M or R is ***false*** in M, written as $\not\models_M R$, iff $v_f^{IA}(R) =$ ***f*** for any assignment ***A***.

Model Theory for CFOPC: Validity of Formula

♣ ***Validity*** of a formula

- For the first order language L(**C**, **F**, **P**) and any $R \in$ **WFF**, R is ***valid***, written as $\models_{\mathbf{CFOPC}} R$, iff $\models_M R$ in any model M for the language.
- Ex: $(A \vee \neg A)$

♣ ***Unsatisfiability*** of a formula

- For the first order language L(**C**, **F**, **P**) and any $R \in$ **WFF**, R is ***unsatisfiable***, written as $\not\models_{\mathbf{CFOPC}} R$, iff $\not\models_M R$ in any model M for the language.
- Ex.: $(A \wedge \neg A)$

♣ ***The undecidability of CFOPC*** [A. Church, 1936, A. M. Turing, 1936]

- Theorem (***The undecidability of CFOPC***): The validity problem for **CFOPC**, i.e., whether a formula of **CFOPC** is valid or not, is undecidable.
- Note: The undecidability of **CFOPC** is one of the fundamental results for logic as well as for computer science.

Model Theory for CFOPC: Tautologies, Contradictions, and Contingencies

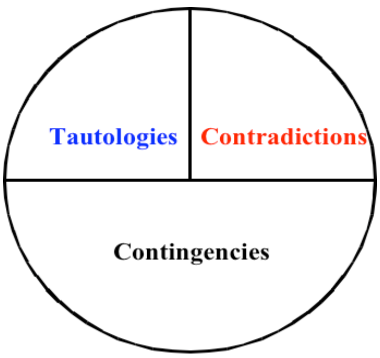
♣ Tautologies, contradictions, and contingencies

- A formula $A \in$ **WFF** is a ***tautology*** of **CFOPC**, written as $\models_{\mathbf{CFOPC}} A$, iff $\models_M A$ for any model M of **CFOPC**;
  a formula $A \in$ **WFF** is a ***contradiction*** of **CFOPC**, written as $\not\models_{\mathbf{CFOPC}} A$, iff $\not\models_M A$ for any model M of **CFOPC**;
  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 **CFOPC** is denoted by **Th**(**CFOPC**).

♣ Relationship between tautologies and contradictions

- Theorem: For any $A \in$ **WFF**, A is a tautology iff $(\neg A)$ is a contradiction, and A is a contradiction iff $(\neg A)$ is a tautology.

Model Theory for CFOPC: Tautologies, Contradictions, and Contingencies



Model Theory for CFOPC: Models of Formulas

♣ Satisfiability of formulas

- For any model $M = (D, I)$ of the first order language L(**C**, **F**, **P**) and any $\Gamma \subseteq$ **WFF**, Γ is ***satisfiable*** in M if there is some assignment ***A*** (called a ***satisfying assignment***) such that under ***A***, $v_f^{IA}(R) =$ ***t*** for all $R \in \Gamma$.
- Theorem (***Compactness***): Let Γ be a set of sentences. If every finite subset of Γ is satisfiable in model M, so is Γ.

♣ Models of formulas

- For any model $M = (D, I)$ of the first order language L(**C**, **F**, **P**) and any $\Gamma \subseteq$ **WFF**, M is called a ***model*** of Γ iff $\models_M R$ (i.e., $v_f^{IA}(R) =$ ***t*** for any assignment ***A***) for any $R \in \Gamma$.
- The set of all models of Γ is denoted by ***M***(***Γ***).

♣ ***Consistence*** of formulas

- For any $\Gamma \subseteq$ **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.

An Introduction to Classical Predicate 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 $\Gamma \models_{\mathbf{CFOPC}} A$, 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 $\boldsymbol{C}_{sem}(\Gamma)$.
  - $\models_{\mathbf{CFOPC}} A =_{df} \phi \models_{\mathbf{CFOPC}} A$ and it means that $\models_M A$ for any model M, i.e., A is a tautology.
- ♣ Note
  - The semantic (model-theoretical or logical) consequence relation of **CFOPC** 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 **CFOPC** iff both $\{A\} \models_{\mathbf{CFOPC}} B$ and $\{B\} \models_{\mathbf{CFOPC}} A$.
  - Theorem: A is semantically (model-theoretically or logically) equivalent to B iff $(A \leftrightarrow B)$ is a tautology.
- ♣ Properties of semantic (model-theoretical or logical) consequence relation
  - The same as those of **CPC**.

Semantic Deduction Theorems

- ♣ Semantic deduction theorems
  - ***Semantic (model-theoretical) deduction theorem for CFOPC***: For any $A, B \in$ **WFF** and any $\Gamma \subseteq$ **WFF**,
    $$\Gamma \cup \{A\} \models_{\mathbf{CFOPC}} B \text{ iff } \Gamma \models_{\mathbf{CFOPC}} (A \rightarrow B).$$
  - ***Semantic (model-theoretical) deduction theorem for CFOPC for finite consequences***: For any $A_1, ..., A_{n-1}, A_n, B \in$ **WFF** and any $\Gamma \subseteq$ **WFF**,
    $$\Gamma \cup \{A_1, ..., A_{n-1}, A_n\} \models_{\mathbf{CFOPC}} B \text{ iff } \Gamma \models_{\mathbf{CFOPC}} (A_1 \rightarrow (...(A_{n-1} \rightarrow (A_n \rightarrow B))...));$$
    $$\Gamma \cup \{A_1, ..., A_{n-1}, A_n\} \models_{\mathbf{CFOPC}} B \text{ iff } \Gamma \models_{\mathbf{CFOPC}} ((A_1 \wedge (...(A_{n-1} \wedge A_n)...)) \rightarrow B).$$
- ♣ Notes
  - As a special case of the above deduction theorems, $\{A\} \models_{\mathbf{CFOPC}} B$ iff $\models_{\mathbf{CFOPC}} (A \rightarrow B)$, i.e., A semantically (model-theoretically or logically) entails B iff $(A \rightarrow B)$ is a tautology.
  - In the framework of **CFOPC**, 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 Predicate Calculus

Uniform Notation of First Order Formulas

- ♣ Uniform notation of first order formulas [R. M. Smullyan, 1968]
  - Classify all quantified formulas and their negations into two categories, i.e., ***γ-formulas*** which act universally, and ***δ-formulas***, which act existentially.
  - For each variety and for each term t, an instance is defined.
- ♣ Proposition
  - Let S be a set of sentences (closed formulas), and γ and δ be sentences. If $S \cup \{\gamma\}$ is satisfiable, so is $S \cup \{\gamma, \gamma(t)\}$ for any closed term t. If $S \cup \{\delta\}$ is satisfiable, so is $S \cup \{\delta, \delta(p)\}$ for any constant symbol p that is new to S and δ.

Uniform Notation of First Order Formulas

♣ γ-formulas and δ-formulas and their instances

Universal		Existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$(\forall x\Phi)$	$\Phi[x/t]$	$(\exists x\Phi)$	$\Phi[x/t]$
$\neg(\exists x\Phi)$	$\neg\Phi[x/t]$	$\neg(\forall x\Phi)$	$\neg\Phi[x/t]$

L: A Hilbert Style Formal System for CFOPC

♣ Axiom schemata of **L**

- $(A\rightarrow(B\rightarrow A))$
  $((A\rightarrow(B\rightarrow C))\rightarrow((A\rightarrow B)\rightarrow(A\rightarrow C)))$
  $(((\neg A)\rightarrow(\neg B))\rightarrow(B\rightarrow A))$
  $((\forall xA)\rightarrow A)$, if x does not occur free in A
  $((\forall xA)\rightarrow A[x/t])$, if x may appear free in A and t is free for x in A
  (i.e., free variables of t do not occur bound in A)
  $((\forall x(A\rightarrow B))\rightarrow(A\rightarrow(\forall xB)))$, if x does not occur free in A

♣ Inference rules of **L**

- Modus Ponens for material implication: from A and $(A\rightarrow B)$ to infer B.
- ***Generalization***: From A to infer $(\forall xA)$.

Properties of L

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

- For any $A, B \in$ **WFF** and any $\Gamma \subseteq$ **WFF**, if $\Gamma\cup\{A\} \vdash_L B$, and no use was made of generalization involving a free variable of A, then $\Gamma \vdash_L A\rightarrow B$.

♣ ***Soundness theorems for L***

- Theorem (***soundness***): If $\vdash_L A$ then $\models_{\mathbf{CFOPC}} A$, for any $A \in$ **WFF**.
- Theorem (***strong soundness***): If $\Gamma \vdash_L A$ then $\Gamma \models_{\mathbf{CFOPC}} A$, for any $A \in$ **WFF** and any $\Gamma \subseteq$ **WFF**.

♣ ***Completeness theorems for L***

- Theorem (***completeness***): If $\models_{\mathbf{CFOPC}} A$ then $\vdash_L A$, for any $A \in$ **WFF**.
- Theorem (***strong completeness***): If $\Gamma \models_{\mathbf{CFOPC}} A$ then $\Gamma \vdash_L A$, for any $A \in$ **WFF** and any $\Gamma \subseteq$ **WFF**.

♣ **CFOPC vs. L**

- **Th**(**CFOPC**) = **Th**(**L**).

An Introduction to Classical Predicate Calculus

♣ The Limitations of Propositional Logic **CPC**
♣ Formal (Object) Language of Classical First Order Predicate Calculus (**CFOPC**)
♣ Substitutions
♣ Model Theory for **CFOPC**
♣ Semantic (Model-theoretical or Logical) Consequence Relation
♣ Hilbert Style Formal Systems for **CFOPC**
♣ Gentzen's Natural Deduction System for **CFOPC**
♣ Gentzen's Sequent Calculus System for **CFOPC**
♣ Semantic Tableau Systems for **CFOPC**
♣ Resolution Systems for **CFOPC**
♣ Second Order Predicate Calculus