THE PREDICATE CALCULUS

2.0 Introduction
2.1 The Propositional Calculus
2.2 The Predicate Calculus
2.3 Using Inference Rules to Produce Predicate Calculus Expressions
2.4 Application: A Logic-Based Financial Advisor
2.5 Epilogue and References
2.6 Exercises
DEFINITION

TRUTH VALUE OF PREDICATE CALCULUS EXPRESSIONS

Assume an expression \( E \) and an interpretation \( I \) for \( E \) over a nonempty domain \( D \). The truth value for \( E \) is determined by:

1. The value of a constant is the element of \( D \) it is assigned to by \( I \).
2. The value of a variable is the set of elements of \( D \) it is assigned to by \( I \).
3. The value of a function expression is that element of \( D \) obtained by evaluating the function for the parameter values assigned by the interpretation.
4. The value of truth symbol “true” is \( T \) and “false” is \( F \).
5. The value of an atomic sentence is either \( T \) or \( F \), as determined by the interpretation \( I \).
6. The value of the negation of a sentence is \( T \) if the value of the sentence is \( F \) and is \( F \) if the value of the sentence is \( T \).
7. The value of the conjunction of two sentences is \( T \) if the value of both sentences is \( T \) and is \( F \) otherwise.
8.–10. The truth value of expressions using \( \lor \), \( \rightarrow \), and \( \equiv \) is determined from the value of their operands as defined in Section 2.1.2.

Finally, for a variable \( X \) and a sentence \( S \) containing \( X \):

11. The value of \( \forall X \ S \) is \( T \) if \( S \) is \( T \) for all assignments to \( X \) under \( I \), and it is \( F \) otherwise.
12. The value of \( \exists X \ S \) is \( T \) if there is an assignment to \( X \) in the interpretation under which \( S \) is \( T \); otherwise it is \( F \).
First-Order Predicate Calculus

- Universally and existentially quantified variables may refer only to objects (constants) in the domain of discourse
- Predicate and function names may not be replaced by quantified variables

**Definition**

**First-Order Predicate Calculus**

*First-order predicate calculus* allows quantified variables to refer to objects in the domain of discourse and not to predicates or functions.
Examples

• If it doesn’t rain on Monday, Tom will go to the mountain
  \( \neg \text{weather}(\text{rain}, \text{monday}) \rightarrow \text{go}(\text{tom}, \text{mountain}) \)

• Emma is a Doberman pinscher and a good dog
  \( \text{gooddog}(\text{emamma}) \land \text{isa}(\text{emamma}, \text{doberman}) \)

• All basketball players are tall
  \( \forall X (\text{basketball\_player}(X) \rightarrow \text{tall}(X)) \)
Examples continued

• Some people like anchovies
  \[ \exists X \ (\text{person}(X) \land \text{likes}(X, \text{anchovies})) \]

• If wishes were horses, beggars would ride
  \[ \text{equal}(\text{wishes}, \text{horses}) \rightarrow \text{ride}(\text{beggars}) \]

• Nobody likes taxes
  \[ \neg \exists X \ \text{likes}(X, \text{taxes}) \]
Example of Semantic Meaning
Blocks World

- Does a given block have a clear top surface?
- Can we pick up block “a”? etc…

- Need to create a set of predicate calculus expressions that represent a static snapshot of the blocks world problem domain
- The set of blocks offers an interpretation and a possible model for the set of predicate calculus expressions
\[
\begin{align*}
on(c,a) \\
on(b,d) \\
onetable(a) \\
onetable(d) \\
clear(b) \\
clear(c) \\
hand\_empty
\end{align*}
\]
• The program needs to infer that block “a” has become clear:
• the following rule describes when a block is clear

\[ \forall X (\neg \exists Y \text{ on}(Y, X) \rightarrow \text{clear}(X)) \]

• that is, for all X, X is clear
  if there does not exist a Y such that Y is on X

• stacking one block on top of another

\[ \forall X \forall Y ((\text{hand}_\text{empty} \land \text{clear}(X) \land \text{clear}(Y) \land \text{pick}_\text{up}(X) \land \text{put}_\text{down}(X, Y)) \rightarrow \text{stack}(X, Y)) \]
Inference Rules

- Semantics of predicate calculus provides a basis for a formal theory of logical inference
- infer new expressions from a set of true assertions
- these expressions are consistent with all previous interpretations of the original set of expressions
Informally,

- An interpretation that makes a sentence true is said to **satisfy** that sentence.

- An interpretation that satisfies every member of a set of expressions is said to **satisfy** the set.

- An expression **X logically follows** from a set of predicate calculus expressions **S** if every interpretation that satisfies **S** also satisfies **X**.
Inference rules

- Inference rules provide a computationally feasible way to determine when an expression, a component of an interpretation, logically follows for that interpretation.
- Inference rules produce new sentences based on the syntactic form of given logical assertions.
- When every sentence $X$ produced by an inference rule operating on a set $S$ of logically expressions logically follows from $S$, the inference rule is said to be **sound**.
- If the inference rule is able to produce every expression that logically follows from $S$ is said to be **complete**.
**DEFINITION**

**SATISFY, MODEL, VALID, INCONSISTENT**

For a predicate calculus expression \( X \) and an interpretation \( I \):

- If \( X \) has a value of \( T \) under \( I \) and a particular variable assignment, then \( I \) is said to satisfy \( X \).

- If \( I \) satisfies \( X \) for all variable assignments, then \( I \) is a *model* of \( X \).

- \( X \) is *satisfiable* if and only if there exist an interpretation and variable assignment that satisfy it; otherwise, it is *unsatisfiable*.

- A set of expressions is *satisfiable* if and only if there exist an interpretation and variable assignment that satisfy every element.

- If a set of expressions is not satisfiable, it is said to be *inconsistent*.

- If \( X \) has a value \( T \) for all possible interpretations, \( X \) is said to be *valid*. 
DEFINITION

PROOF PROCEDURE

A *proof procedure* is a combination of an inference rule and an algorithm for applying that rule to a set of logical expressions to generate new sentences.

We present proof procedures for the *resolution* inference rule in Chapter 11.
**DEFINITION**

LOGICALLY FOLLOWS, SOUND, and COMPLETE

A predicate calculus expression **X** *logically follows* from a set **S** of predicate calculus expressions if every interpretation and variable assignment that satisfies **S** also satisfies **X**.

An inference rule is *sound* if every predicate calculus expression produced by the rule from a set **S** of predicate calculus expressions also logically follows from **S**.

An inference rule is *complete* if, given a set **S** of predicate calculus expressions, the rule can infer every expression that logically follows from **S**.
DEFINITION

MODUS PONENTS, MODUS TOLLENS, AND ELIMINATION, AND INTRODUCTION, and UNIVERSAL INSTANTIATION

If the sentences $P$ and $P \rightarrow Q$ are known to be true, then *modus ponens* lets us infer $Q$.

Under the inference rule *modus tollens*, if $P \rightarrow Q$ is known to be true and $Q$ is known to be false, we can infer $\neg P$.

*And elimination* allows us to infer the truth of either of the conjuncts from the truth of a conjunctive sentence. For instance, $P \land Q$ lets us conclude $P$ and $Q$ are true.

*And introduction* lets us infer the truth of a conjunction from the truth of its conjuncts. For instance, if $P$ and $Q$ are true, then $P \land Q$ is true.

*Universal instantiation* states that if any universally quantified variable in a true sentence is replaced by any appropriate term from the domain, the result is a true sentence. Thus, if $a$ is from the domain of $X$, $\forall X \ p(X)$ lets us infer $p(a)$. 
Examples of the use of modus ponens

• In the propositional calculus
  “if it is raining, then the ground will be wet” and
  “it is raining”
  if $P$ denotes “it is raining”
  $Q$ denotes “the ground is wet”

• The first expression becomes $P \rightarrow Q$

• Our set of axioms becomes
  
  $P \rightarrow Q$

• After applying modus ponens, the fact that the
  ground is wet ($Q$) may be added to the set of true
  expressions
Examples continued

• “All men are mortal” and “Socrates is a man
  \( \forall X (\text{man}(X) \rightarrow \text{mortal}(X)) \)
  \text{man(socrates)}

• under inference rule of universal instantiation
  \text{man(socrates)} \rightarrow \text{mortal(socrates)}

• applying modus ponens, infer the conclusion
  \text{mortal(socrates)}

• This is added to the set of expressions that logically follow from the original assertions
Unification

- An algorithm for determining the substitutions needed to make two predicate calculus expressions match

- In the previous example, *socrates* was substituted for *X* in

\[ \forall X (\text{man}(X) \rightarrow \text{mortal}(X)) \]

then it becomes \( \text{man}(\text{socrates}) \)

after applying modus ponens, the conclusion is

\( \text{mortal}(\text{socrates}) \)

- If \( P(X) \) and \( P(Y) \) are equivalent, \( Y \) may be substituted for \( X \) to make the sentences match
Essential Aspect of Unification

- The requirement is that all variables be universally quantified
- this allows full freedom in computing substitutions
- existential quantified variables may be eliminated from sentences in the database by replacing them with constants that make the sentence true
- $\exists X \text{parent}(X, \text{tom})$ could be replaced by the expressions $\text{parent}(\text{bob}, \text{tom})$ or $\text{parent}(\text{mary}, \text{tom})$ assuming that bob and mary are tom’s parents under interpretation
- $\forall X \text{mother}(X, f(X))$ where Y depends on X
- $\forall X \exists Y \text{mother}(X, Y)$ Y is replaced by a skolem function
Substitution

• For some instances,

\[ \text{foo}(X,a,\text{goo}(Y)) \] can be generated by the following legal substitutions

\[ \text{foo}(\text{fred},a,\text{goo}(Z)) \]
\[ \text{foo}(W,a,\text{goo}(\text{jack})) \]
\[ \text{foo}(Z,a,\text{goo}(\text{moo}(Z))) \]

• substitution instance or unification can be seen as

\[ \{\text{fred}/X, Z/Y\} \]
\[ \{W/X, \text{jack}/Y\} \]
\[ \{Z/X, \text{moo}(Z)/Y\} \]
Substitution continued

• X/Y indicates that X is substituted for the variable Y in the original expression
• in \{fred/X, Z/Y\}, fred is substituted for X and Z is substituted for Y

Binding
• substitutions are also referred as bindings.
• a variable is said to be bound to the value substituted for it.
Substitution continued

• substitution required to match two expressions
• a constant may be substituted for a variable
• any constant is considered ground instance and may not be replaced
• neither can two different ground instances be substituted for one variable
• a variable cannot be unified with a term containing that variable, i.e. \( X \) cannot be replaced by \( p(X) \)
• logic problem solvers must maintain consistency of variable substitutions
Substitution continued

- Any unifying substitution be made consistently across all occurrences of the variable in both expressions being matched

Example

- socrates was substituted not only for the variable X in man(X), but also for the variable X in mortal(X)
Substitution continued

• Once a variable has been bound, future unifications and inferences must take the value of this binding into account

• If variable $X_1$ is substituted for another variable $X_2$ and at a later time $X_1$ is replaced by a constant, then $X_2$ must also reflect this binding

Example

• if $p(a,X)$ unifies with the premise of $p(Y,Z) \Rightarrow q(Y,Z)$ with the substitution $\{a/Y,X/Z\}$, we can infer $q(a,X)$ under the same substitution

• match with the premise of $q(W,b) \Rightarrow r(W,b)$, we infer $r(a,b)$ under the substitution set $\{a/W,b/X\}$