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

George F. Luger

ARTIFICIAL INTELLIGENCE
Structure and Strategies for Complex Problem Solving

Fourth Edition
DEFINITION

PROPOSITIONAL CALCULUS SYMBOLS

The *symbols* of propositional calculus are the propositional symbols:

P, Q, R, S, ...

truth symbols:

true, false

and connectives:

∧, ∨, ¬, →, ≡
DEFINITION

PROPOSITIONAL CALCULUS SENTENCES

Every propositional symbol and truth symbol is a sentence.

For example: true, P, Q, and R are sentences.

The negation of a sentence is a sentence.

For example: ¬P and ¬false are sentences.

The conjunction, or and, of two sentences is a sentence.

For example: P ∧ ¬P is a sentence.

The disjunction, or or, of two sentences is a sentence.

For example: P ∨ ¬P is a sentence.

The implication of one sentence from another is a sentence.

For example: P → Q is a sentence.

The equivalence of two sentences is a sentence.

For example: P ∨ Q ⊨ R is a sentence.

Legal sentences are also called well-formed formulas or WFFs.
DEFINITION

PROPOSITIONAL CALCULUS SEMANTICS

An interpretation of a set of propositions is the assignment of a truth value, either $\text{T}$ or $\text{F}$, to each propositional symbol.

The symbol $\text{true}$ is always assigned $\text{T}$, and the symbol $\text{false}$ is assigned $\text{F}$.

The interpretation or truth value for sentences is determined by:

The truth assignment of negation, $\neg P$, where $P$ is any propositional symbol, is $\text{F}$ if the assignment to $P$ is $\text{T}$, and $\text{T}$ if the assignment to $P$ is $\text{F}$.

The truth assignment of conjunction, $\land$, is $\text{T}$ only when both conjuncts have truth value $\text{T}$; otherwise it is $\text{F}$.

The truth assignment of disjunction, $\lor$, is $\text{F}$ only when both disjuncts have truth value $\text{F}$; otherwise it is $\text{T}$.

The truth assignment of implication, $\rightarrow$, is $\text{F}$ only when the premise or symbol before the implication is $\text{T}$ and the truth value of the consequent or symbol after the implication is $\text{F}$; otherwise it is $\text{T}$.

The truth assignment of equivalence, $\equiv$, is $\text{T}$ only when both expressions have the same truth assignment for all possible interpretations; otherwise it is $\text{F}$. 
For propositional expressions $P$, $Q$ and $R$:

$\neg (\neg P) \equiv P$

$(P \lor Q) \equiv (\neg P \rightarrow Q)$

the contrapositive law: $(P \rightarrow Q) \equiv (\neg Q \rightarrow \neg P)$

de Morgan’s law: $\neg (P \lor Q) \equiv (\neg P \land \neg Q)$ and $\neg (P \land Q) \equiv (\neg P \lor \neg Q)$

the commutative laws: $(P \land Q) \equiv (Q \land P)$ and $(P \lor Q) \equiv (Q \lor P)$

the associative law: $((P \land Q) \land R) \equiv (P \land (Q \land R))$

the associative law: $((P \lor Q) \lor R) \equiv (P \lor (Q \lor R))$

the distributive law: $P \lor (Q \land R) \equiv (P \lor Q) \land (P \lor R)$

the distributive law: $P \land (Q \lor R) \equiv (P \land Q) \lor (P \land R)$
**Figure 2.1:** Truth table for the operator $\wedge$.

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>$P \wedge Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Figure 2.2: Truth table demonstrating the equivalence of $P \not\equiv Q$ and $\square P \triangle Q$.

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\neg P$</th>
<th>$\neg P \lor Q$</th>
<th>$P \Rightarrow Q$</th>
<th>$(\neg P \lor Q) = (P \Rightarrow Q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
The Predicate Calculus

- access the components of an individual assertion
- allow expression to contain variables

Example
- single proposition
  \[ P = \text{it rained on Tuesday} \]

- Using predicate `weather`
  \[ \text{weather(tuesday, rain)} \]
The Predicate Calculus Continued

- general assertion about clauses of entities

for all values of X, where X is a day of the week,

\( \text{weather}(X,\text{rain}) \)

- means that it rains every day
**DEFINITION**

**PREDICATE CALCULUS SYMBOLS**

The alphabet that makes up the symbols of the predicate calculus consists of:

1. The set of letters, both upper- and lowercase, of the English alphabet.
2. The set of digits, 0, 1, …, 9.
3. The underscore, _.

*Symbols* in the predicate calculus begin with a letter and are followed by any sequence of these legal characters.

Legitimate characters in the alphabet of predicate calculus symbols include

```plaintext
a R 6 9 p _ z
```

Examples of characters not in the alphabet include

```plaintext
# % @ / & “ ”
```

Legitimate predicate calculus symbols include

```plaintext
George fire3 tom_and_jerry bill XXXX friends_of
```

Examples of strings that are not legal symbols are

```plaintext
3jack “no blanks allowed” ab%cd ***71 duck!!
```
**DEFINITION**

**SYMBOLS and TERMS**

Predicate calculus symbols include:

1. *Truth symbols* **true** and **false** (these are reserved symbols).
2. *Constant symbols* are symbol expressions having the first character lowercase.
3. *Variable symbols* are symbol expressions beginning with an uppercase character.
4. *Function symbols* are symbol expressions having the first character lowercase. Functions have an attached arity indicating the number of elements of the domain mapped onto each element of the range.

A *function expression* consists of a function constant of arity **n**, followed by **n** terms, \( t_1, t_2, \ldots, t_n \), enclosed in parentheses and separated by commas.

A predicate calculus *term* is either a constant, variable, or function expression.
Examples

• Well-formed function expressions
  
  \[ f(X, Y) \]
  
  father(david)
  
  price(bananas)

• arity (argument)
  
  in father(david), arity is 1
  
  in plus(2,3), arity is 2
DEFINITION

PREDICATES and ATOMIC SENTENCES

Predicate symbols are symbols beginning with a lowercase letter.

Predicates have an associated positive integer referred to as the \textit{arity} or “argument number” for the predicate. Predicates with the same name but different arities are considered distinct.

An atomic sentence is a predicate constant of arity $n$, followed by $n$ terms, $t_1, t_2, \cdots, t_n$, enclosed in parentheses and separated by commas.

The truth values, \textbf{true} and \textbf{false}, are also atomic sentences.
Examples of atomic sentences

- A predicate relation is defined by its name and its arity
  
  likes(george,kate)
  likes(george,sarah,tuesday)
  friends(bill,george)
  friends(father_of(david),father_of(andrew))
  helps(richard,bill)
• atomic sentences are also called atomic expressions, atoms, or propositions

• combine atomic sentences using logical operators to form sentences in the predicate calculus

• operators are the same as connectives in propositional calculus

\[ \land \quad \lor \quad \neg \quad \rightarrow \quad \equiv \]
Quantifiers

- Universal quantifier \( \forall \)

  is true for all values in the domain of the definition of \( X \)

- Variable quantifier \( \exists \)

  \[ \exists Y \text{ friends}(Y, \text{peter}) \]

  is true if there is at least one object, indicated by \( Y \) that is a friend of peter
DEFINITION

PREDICATE CALCULUS SENTENCES

Every atomic sentence is a sentence.

1. If \( s \) is a sentence, then so is its negation, \( \neg s \).
2. If \( s_1 \) and \( s_2 \) are sentences, then so is their conjunction, \( s_1 \land s_2 \).
3. If \( s_1 \) and \( s_2 \) are sentences, then so is their disjunction, \( s_1 \lor s_2 \).
4. If \( s_1 \) and \( s_2 \) are sentences, then so is their implication, \( s_1 \rightarrow s_2 \).
5. If \( s_1 \) and \( s_2 \) are sentences, then so is their equivalence, \( s_1 \equiv s_2 \).
6. If \( X \) is a variable and \( s \) a sentence, then \( \forall X \ s \) is a sentence.
7. If \( X \) is a variable and \( s \) a sentence, then \( \exists X \ s \) is a sentence.
Examples

- let times and plus be functions symbols of arity 2 and
- let equal and foo be predicate symbols with arity 2 and 3 respectively

plus(two,three) is a function, not an atomic sentence

equal(plus(2,3),five) is an atomic sentence

equal(plus(2,3),7) is an atomic sentence

$\exists X \ foo(X,\text{two},plus(2,3)) \land equal(plus(2,3),5)$
verify_sentence algorithm

function verify_sentence(expression);
begin
  case
    expression is an atomic sentence: return SUCCESS;
    expression is of the form Q X s, where Q is either \( \forall \) or \( \exists \), X is a variable, and s is an expression;
      if verify_sentence(s) returns SUCCESS
      then return SUCCESS
      else return FAIL;
    expression is of the form \( \neg s \):
      if verify_sentence(s) returns SUCCESS
      then return SUCCESS
      else return FAIL;
    expression is of the form \( s_1 \ op \ s_2 \), where op is a binary logical operator:
      if verify_sentence(s_1) returns SUCCESS and
      verify_sentence(s_2) returns SUCCESS
      then return SUCCESS
      else return FAIL;
    otherwise: return FAIL
  end
end.
Use of predicate calculus

mother(eve, abel)
father(adam, abel)
mother(eve, cain)
father(adam, cain)

\( \forall X \forall Y (father(X,Y) \lor mother(X,Y) \rightarrow parent(X,Y)) \)

\( \forall X \forall Y \forall Z (parent(X,Y) \land parent(X,Z) \rightarrow sibling(Y,Z)) \)
Semantics for the Predicate Calculus

- It is important to determine well-formed expressions’ meaning in terms of objects, predicates and relations in the world.

- To use the predicate calculus as a representation for problem solving, we need to describe objects and relations in the domain of interpretation with a set of well-formed expressions.

- The terms and predicates of these expressions denote objects and relations in the domain.

- The database of predicate calculus expressions, each having truth value T, describes as “state of the world.”
DEFINITION

INTERPRETATION

Let the domain $D$ be a nonempty set.

An *interpretation* over $D$ is an assignment of the entities of $D$ to each of the constant, variable, predicate, and function symbols of a predicate calculus expression, such that:

1. Each constant is assigned an element of $D$.
2. Each variable is assigned to a nonempty subset of $D$; these are the allowable substitutions for that variable.
3. Each function $f$ of arity $m$ is defined on $m$ arguments of $D$ and defines a mapping from $D^m$ into $D$.
4. Each predicate $p$ of arity $n$ is defined on $n$ arguments from $D$ and defines a mapping from $D^n$ into $\{T, F\}$. 
Quantification of Variables

• likes(george,X)
  substituting kate and susie for X in likes(george, X)
  becomes
  likes(george,kate) and likes(george,susie)
• Variable X can be replaced to Y without changing meaning, and so called as “dummy”
• Variables must be quantified in either universally or existentially
• A variable is considered “free” if it is not within the scope of either the universal or existential quantifiers
• An expression is “closed” if all of its variables are quantified
• A “ground expression” has no variables at all
• In Predicate Calculus, all variables must be quantified
• Scope of quantification with parenthesis, $X$ is universally quantified in both $p(X)$ and $r(X)$

$$\forall X (p(X) \lor q(Y) \rightarrow r(X))$$

• Relationship between negation and universal and existential quantifiers

$$\neg \exists X \ p(X) \equiv \forall X \ \neg p(X)$$
$$\neg \forall X \ p(X) \equiv \exists X \ \neg p(X)$$
$$\exists X \ p(X) \equiv \exists X \ p(Y)$$
$$\forall X \ q(X) \equiv \forall Y \ q(Y)$$
$$\forall X \ (p(X) \land q(X)) \equiv \forall X \ p(X) \land \forall Y \ q(Y)$$
$$\exists X \ (p(X) \lor q(X)) \equiv \exists X \ p(X) \lor \exists Y \ q(Y)$$