Predicate Calculus - Semantics

Lila Kari

University of Waterloo
The first-order language $\mathcal{L}^{\text{pred}}$, even if associated with a structure (set theory, number theory) is a syntactic object of no semantic significance. The formulas of $\mathcal{L}^{\text{pred}}$, however, are intended to express statements. This is accomplished by interpretations.

Interpretations for the formulas in the propositional language $\mathcal{L}^p$ were simple: they consisted of value assignments of the propositional symbols.

The language $\mathcal{L}^{\text{pred}}$ includes more classes of symbols and hence the interpretations for it are more complex.

An interpretation for a statement in $\mathcal{L}^{\text{pred}}$ must contain sufficient information to determine whether the statement is true or false.
Interpretations

1. The logical symbols:

- Of the logical symbols, the connectives will be interpreted as in Propositional Logic.
- The meaning of quantifiers has been explained intuitively.
- The equality symbol ≈ denotes the relation of equality.
- The variable symbols will be interpreted as variables ranging over the domain.
- Punctuation symbols serve just like punctuation in natural languages.
2. The non-logical symbols (parameters):

- In the case when $\mathcal{L}^{pred}$ is associated with a structure, they are interpreted as the designated individuals, relations, and functions related with the structure. Accordingly, the sentences (closed formulas) are intended to express propositions about the structure.

- If $\mathcal{L}^{pred}$ is not associated with any structure, a domain is still required for each interpretation. In such cases, however, the domain is merely an arbitrary non-empty set. Then the non-logical symbols are interpreted as arbitrary individuals, and arbitrary relations and functions on it.
• Recall that an $n$-ary relation on a domain $D$ is a subset of $D^n$

• Recall that the equality relation on $D$ is the subset \{(x, y) | x, y \in D \text{ and } x = y\} or alternatively \{(x, x) | x \in D\}.

• The italic capital Latin letter $I$ (with or without subscripts and superscripts) will be used for any interpretation.
An **interpretation** \( I \) for the first-order language \( \mathcal{L}^{\text{pred}} \) consists of a domain \( D \) called the domain of the interpretation, and a function (denoted also by \( I \)) with the properties:

1. For each individual constant symbol \( a \), we have that \( I(a) \in D \).
2. For each relation symbol (predicate) \( P \), \( I(P) \) is an \( n \)-ary relation on \( D \), that is, \( I(P) \subseteq D^n \).
3. For each \( m \)-ary function symbol \( f \), \( I(f) \) is a total \( m \)-ary function of \( D \) into \( D \), that is, \( I(f) : D^m \rightarrow D \).

**Note:** \( I(a) \), \( I(P) \), \( I(f) \) should not be confused with \( a \), \( P \), \( f \).
Proof that Unicorns Exist

*(R. Smullyan, based on an argument by Descartes)*

- Prove the (possibly) stronger statement that there exists an existing unicorn (an existing unicorn is, of course, an unicorn that exists).
- There are exactly two possibilities:
  1. An existing unicorn exists.
  2. An existing unicorn does not exist.
- Possibility (2) clearly is contradictory.
- Thus, only possibility (1) can occur, q.e.d.
What is wrong with the proof?

• The fallacy lies in the double-meaning of the word *an*, which in some contexts means **every** (\(\forall\)) and in others means **at least one** (\(\exists\)).
  
  • “*An* owl has large eyes” means **All owls have large eyes**.
  • “*An* owl is in my house” means **There exists an owl in my house**.

• When we say “*An* existing unicorn exists” it is not clear if we mean **All existing unicorns exist** or **There exists at least one existing unicorn**.
  
  • If we mean the **first**, then it is true (how can we have an existing unicorn that does not exist?)
  • But this does not mean that the statement is true in the **second** sense, that is, that there must exist at least an existing unicorn.
The importance of a non-empty domain

• Note that \( \forall x P(x) \) is always vacuously true, when the domain is the empty set.
• This was the case in the example with the unicorns (the first interpretation of the word “an”)
• However, in the same example, \( \exists x P(x) \) was false (the second interpretation of the word “an”)
• In general, we want \( \forall x P(x) \rightarrow \exists x P(x) \) to be true.
• This would be false in an empty domain.
• This is why we defined a domain as being a non-empty set.
Comments on interpretations

As an example - not correct in every detail - think of a programming language function (in the sense of, say, PASCAL).

- The specification of such a function will make a statement about the connection between the input parameters and the function value.
- There may be identifiers declared as constants and there may be global variables.
- To understand what the function does we need to know what domain we are talking about, which values the constants have and which values the global variables have.
- In our context, the global variables are modelled by free variable symbols.
Examples

Consider the sentence (closed formula, no free variables)

$$\forall x (F(x) \lor H(x) \rightarrow G(x)).$$

In one interpretation,

- The domain is $D_1 = \text{the set of all ships}$
- $I_1(F)$ is a unary predicate on $D_1$ defined by $I_1(F)(x)$: $x$ is on fire, which takes the value 1 if $x$ is on fire and the value 0 if $x$ is not on fire.

The values of $I_1(F)$ are known for every ship. This means that for every $x$ in $D_1$ we know whether the statement “$x$ is on fire“ is true or false. $I_1(F)$ corresponds to the subset of $D$ consisting of all those $x$ for which “$x$ is on fire“ is true. This is the subset consisting of all ships that are on fire.
Example contd.

\[ \forall x (F(x) \lor H(x) \rightarrow G(x)). \]

- \( I_1(H) \) is the unary predicate on \( D_1 \) defined by \( I_1(H)(x) : x \ \text{has a hole.} \)
- \( I_1(G) \) is the unary predicate defined by \( I_1(G)(x) : x \ \text{sinks.} \)

In this interpretation the formula says Every ship that is on fire or has a hole sinks.
Another interpretation

\[ \forall x (F(x) \lor H(x) \rightarrow G(x)). \]

In the second interpretation,

- The domain is \( D_2 \) is the set of integers.
- \( I_2(F) \) is the unary predicate defined by \( I_2(F)(x) : x \) is positive.
- \( I_2(H) \) is the unary predicate defined by \( I_2(H)(x) : x \) equals zero.
- \( I_2(G) \) is the predicate defined by \( I_2(G)(x) : x \geq 0. \)

Then the formula says An integer that is 0 or strictly greater than 0 is greater than or equal to 0.
The third interpretation

\[ \forall x (F(x) \lor H(x) \rightarrow G(x)). \]

In a 3rd interpretation, the domain is \( D_3 \) and the function is \( I_3 \) where:

- \( D_3 \) is the set of all animals.
- \( I_3(F) \) is the unary predicate defined by \( x \) is an unicorn.
- \( I_3(H) \) is the unary predicate defined by \( x \) is an rhinoceros.
- \( I_3(G) \) is the unary predicate defined by \( x \) is an animal with exactly one horn.

Then the formula says Every animal which is a unicorn or a rhinoceros has exactly one horn.
A fourth interpretation

\[ \forall x \left( F(x) \lor H(x) \rightarrow G(x) \right). \]

In a 4th interpretation, the domain is \( D_4 \) and the function is \( I_4 \) where:

- \( D_4 \) is the set of all attempted \( \text{C++} \) programs.
- \( I_4(F) \) is the predicate defined by \( I_4(F)(x) \): \( x \) is a syntactically incorrect attempted \( \text{C++} \) program.
- \( I_4(H) \) is the predicate defined by \( I_4(H)(x) \): \( x \) is a semantically incorrect attempted \( \text{C++} \) program.
- \( I_4(G) \) is the predicate defined by \( I_4(G)(x) \): \( x \) is an incorrect attempted \( \text{C++} \) program.

Then the formula says that Every attempted \( \text{C++} \) program which is incorrect syntactically or semantically is incorrect.
• Define the predicate “x loves y” by $L(x, y)$, which is 1 if $x$ loves $y$, and false otherwise.
• English sentences like “everyone loves someone” can be formalized by first-order logic formulas like $\forall x \exists y L(x, y)$.
• Using just the two quantifiers and the predicate $L$, but no logical connectives and no function or individual constant symbols, formulas with 8 different meanings can be built.
• The following diagrams show interpretations that make each of these formulas true, where the domain consists of exactly five individuals $\{a, b, c, d, e\}$ which can love (vertical axis) and be loved (horizontal axis).
• A small red box at row $x$ and column $y$ indicates that $L(x, y) = 1$. 
1. $\forall x \exists y \mathcal{L}(y, x)$
Everyone is loved by someone

2. $\forall x \exists y \mathcal{L}(x, y)$
Everyone loves someone
3. $\exists x \forall y L(x, y)$
Someone loves everyone

4. $\exists x \forall y L(y, x)$
Someone is loved by everyone
5. $\exists x L(x, x)$
Someone loves him/herself

6. $\forall x L(x, x)$
Everyone loves him/herself
7. $\exists x \exists y L(x, y)$  
   Someone loves someone  
8. $\exists x \exists y L(y, x)$  
   Someone is loved by someone

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. $\forall x \forall y L(x, y)$  
   Everyone loves everyone  
10. $\forall x \forall y L(y, x)$  
    Everyone is loved by everyone

![Diagram](image)
1. $\forall x \exists y L(x,y)$: Everyone is loved by someone.
2. $\forall x \exists y L(x,y)$: Everyone loves someone.
3. $\exists x \forall y L(x,y)$: Someone loves everyone.
4. $\exists x \forall y L(y,x)$: Someone is loved by everyone.
5. $\exists x L(x,x)$: Someone loves him/herself.
6. $\forall x L(x,x)$: Everyone loves him/herself.
7. $\exists x \exists y L(x,y)$: Someone loves someone.
8. $\exists x \exists y L(y,x)$: Someone is loved by someone.
9. $\forall x \forall y L(x,y)$: Everyone loves everyone.
10. $\forall x \forall y L(y,x)$: Everyone is loved by everyone.
• Each interpretation, represented by a logical matrix, satisfies the formulas in its caption in a ”minimal” way, i.e. whitening any red cell in any matrix would make it non-satisfying the corresponding formula.

• What other matrices satisfy formula 1?
  Answer: matrices at 3, 6, and 10 (but not those at 2, 4, 5, and 7).

• What formulas are satisfied by the matrix at 6?
  Answer: formulas 1, 2, 5, 6, 7, and 8 (but not 3, 4, 9, and 10).

• Some formulas “imply” others, e.g. formula 3 ”implies” formula 1, that is, each matrix fulfilling formula 3 also fulfills formula 1, but not vice versa (to be proved later)

• For which formulas is the interpretation that makes it true unique?
  Answer: formulas 9 and 10
Hasse diagram of implications

- Matrix not empty
- No column empty
- Diagonal not empty
- No row empty
- One row full
- Diagonal full
- One column full
- Matrix full
We now define the value of a term and of a formula in $\mathcal{L}^{\text{pred}}$ under a given interpretation $I$.

**Definition (Value of a closed term).**

The value of a closed term under interpretation $I$ over domain $D$ is defined by recursion:

1. If $a$ is an individual constant symbol then $I(a) \in D$.
2. If $t_1, t_2, \ldots, t_n$ are closed terms, then

$$I(f(t_1, t_2, \ldots, t_n)) = I(f)(I(t_1), I(t_2), \ldots, I(t_n)).$$
Definition (Value of a sentence)

The value of sentences under interpretation $I$ over domain $D$ is defined by recursion:

1. $I(F(t_1, \ldots, t_n)) = \begin{cases} 1 & \text{if } (I(t_1), I(t_2), \ldots, I(t_n)) \in I(F) \\ 0 & \text{otherwise} \end{cases}$

2. $I(t_1 \approx t_2) = \begin{cases} 1 & \text{if } I(t_1) = I(t_2), \\ 0 & \text{otherwise} \end{cases}$

where $t_1, t_2, \ldots, t_n$ are closed terms.
Values of sentences (closed formulas)

(2)

\[ I(\neg A) = \begin{cases} 
1 & \text{if } I(A) = 0, \\
0 & \text{otherwise.} 
\end{cases} \]

(3)

\[ I(A \land B) = \begin{cases} 
1 & \text{if } I(A) = I(B) = 1, \\
0 & \text{otherwise.} 
\end{cases} \]

(4)

\[ I(A \lor B) = \begin{cases} 
1 & \text{if } I(A) = 1 \text{ or } I(B) = 1 \\
0 & \text{otherwise.} 
\end{cases} \]

(5)

\[ I(A \rightarrow B) = \begin{cases} 
1 & \text{if } I(A) = 0 \text{ or } I(B) = 1, \\
0 & \text{otherwise.} 
\end{cases} \]
Value of closed formulas (closed terms)

(6)
\[ I(A \leftrightarrow B) = \begin{cases} 1 & \text{if } I(A) = I(B) \\ 0 & \text{otherwise.} \end{cases} \]

(7)
\[ I(\forall x A(x)) = \begin{cases} 1 & \text{if for every } d \in D, \text{ by assigning } d \text{ to } u \text{ we get } I(A(u)) = 1, \\ (u \text{ not occurring in } A(x)) \\ 0 & \text{otherwise.} \end{cases} \]

(8)
\[ I(\exists x A(x)) = \begin{cases} 1 & \text{if for some } d \in D, \text{ by assigning } d \text{ to } u \text{ we get } I(A(u)) = 1, \\ (u \text{ not occurring in } A(x)) \\ 0 & \text{otherwise.} \end{cases} \]
**Theorem**

Suppose \( I \) is an interpretation over a domain \( D \), \( t \) is a closed term, and \( A \in \text{Sent}(\mathcal{L}^{\text{pred}}) \). Then

1. \( I(t) \in D \).
2. \( I(A) \in \{0, 1\} \).

**Proof:** By induction on the complexity of \( t \) and \( A \).
Values of terms and formulas

- The preceding definitions are concerned with closed terms and formulas.
- Non-closed terms and formulas contain free variable symbols which are interpreted as variables ranging over the domain.
- Therefore, a term containing $n$ free variables will be interpreted (over domain $D$) not as an individual in $D$, but as an $n$-ary function on $D$.
- A formula containing $n$ free variable symbols will be interpreted not as truth or falsehood, but as an $n$-ary propositional function on $D$.
- That is, the values of non-closed terms and formulas will depend not only on interpretations, but also on the value assignments to the free variable symbols occurring in them.
• A value assignment within an interpretation $I$ over domain $D$ is an assignment of individuals in $D$ to free variable symbols.

• The italic small Latin letter $s$ (with or without subscripts or superscripts) will be used for any assignment.

• The individual constant assigned to a free variable symbol $u$ by $s$ is written as $s(u)$.

• The value of a term $t$ under interpretation $I$ together with an assignment $s$, and the value of a formula $A$ can be now defined. They are written as $[I, s](t)$ and $[I, s](A)$ respectively.
Definition: Value of terms

The value of terms under interpretation $I$ over domain $D$ and with value assignment $s$ within $I$ is defined by recursion:

1. $[I, s](a) = I(a)$ if $a$ is an individual constant symbol,
2. $[I, s](u) = s(u)$ if $u$ is a free variable symbol,
3. $[I, s](f(t_1, \ldots, t_n)) = I(f)([I, s](t_1), \ldots, [I, s](t_n))$ if $f$ is a function symbol and $t_1, \ldots, t_n$ are terms.

Occasionally, a given value assignment $s$ has to be modified for a few free variable symbols. Let $d \in D$. Then $s_{[u=d]}$ denotes the value assignment which coincides with $s$ on all free variable symbols except $u$ (which is assumed to be distinct) and which assigns $d$ to $u$. 
Value of formulas

The value of formulas under interpretation $I$ over domain $D$ and with value assignment $s$ in $I$ is defined by recursion.

(1) If $A = F(t_1, \ldots, t_n)$ then

$$
[I, s](A) = \begin{cases} 
1 & \text{if } ([I, s](t_1), \ldots, [I, s](t_n)) \in I(F) \\
0 & \text{otherwise}
\end{cases}
$$

If $A = t_1 \approx t_2$

$$
[I, s](A) = \begin{cases} 
1 & \text{if } [I, s](t_1) = [I, s](t_2) \\
0 & \text{otherwise}
\end{cases}
$$

(2) - (6) The same as in the definition for closed formulas except that $s$ is involved.
(7) If $A = \forall x B(x)$, let $u$ be a free variable symbol not occurring in $B(x)$.

$$[I, s](A) = \begin{cases} 1 & \text{if for every } d \in D \text{ one has } [I, s_{[u=d]}](B(u)) = 1. \\ 0 & \text{otherwise} \end{cases}$$

(8) If $A = \exists x B(x)$, let $u$ be a free variable symbol not occurring in $B(x)$.

$$[I, s](A) = \begin{cases} 1 & \text{if for some } d \in D \text{ one has } [I, s_{[u=d]}](B(u)) = 1. \\ 0 & \text{otherwise} \end{cases}$$
Cases (7) and (8) call for some explanations.

- $\forall x B(x)$ and $\exists x B(x)$ are generated from $B(u)$.
- Intuitively, $[I, s](\forall x B(x))$ means that $[I, s](B(u)) = 1$ no matter what individual constant $s(u)$ is.
- We note that for each free variable symbol $v$ other than $u$ occurring in $B(x)$ (if any), $s(v)$ is held constant. To express this precisely, we use $s[u=d]$ and require in Case (7) that, for every $d \in D$, $[I, s[u=d]](B(u)) = 1$.
- Similarly so, for Case (8).
- Hence $s[u=d]$ is a representative of $s$ used for the purpose of evaluating the value of $\forall x B(x)$ and $\exists x B(x)$ under $I$ and $s$. 
Theorem. Suppose \( I \) is an interpretation over domain \( D \) and \( s \) is a value assignment within \( I \). Then

1. \([I, s](t) \in D\) for every term \( t \).
2. \([I, s](A) \in \{0, 1\}\) for every formula \( A \in L^{\text{pred}} \).
Example

Suppose
(1) \( t = f(g(a), f(b, c)) \),
(2) \( t_1 = f(g(u), f(v, c)) \)
(3) \( A = f(g(b), g(u)) \approx g(v) \),
(4) \( B = \forall x \exists y (F(y) \land G(x, y)) \),
(5) \( C = \forall x[H(x) \land G(b, x) \rightarrow \exists y \exists z (F(y) \land F(z) \land x \approx f(y, z))] \)

Suppose \( I \) is an interpretation over domain \( \mathbb{N} \) (the set of natural numbers) and \( s \) is a value assignment within \( I \) such that:
\( I(a) = 1, I(b) = 2, I(c) = 3, s(u) = 4, s(v) = 5 \)
\( I(F)(x) \) means \( x \) is prime
\( I(G)(x, y) \) means \( x < y \)
\( I(H)(x) \) means \( x \) is even
\( I(f)(x, y) = x + y \)
\( I(g)(x) = x^2 \).
Example contd.

Then we have

(1) $I(t) = 1^2 + (2 + 3) = 6$
(2) $[I, s](t_1) = 4^2 + (5 + 3) = 24$
(3) $[I, s](A)$ is $2^2 + 4^2 = 5^2$ which is false,
(4) $[I, s](B)$ is Every natural number is less than some prime number, or There are infinitely many prime numbers which is true (Euclid’s Theorem).
(5) $[I, s](C)$ is Every even number greater than 2 equals the sum of two primes, the value of which has not yet been decided (Goldbach’s conjecture).

It should be noticed that interpretation together with assignments are analogous to, but not the same as, value assignments defined for propositional formulas.
Example

(a) Find the value of the formula

\[ A = \forall x (N(x) \rightarrow \exists y (N(y) \land G(y, x))) \]

under the interpretation \( I \) over the domain \( D = \mathbb{R} \) where

\( I(N)(x) \) means \( x \) is an integer.
\( I(G)(y, x) \) means \( y > x \)

(b) Find the value of the same formula under the interpretation \( I' \) over the domain \( D' = \mathbb{N} \) where

\( I'(N)(x) \) means \( x \) is odd,
\( I'(G)(y, x) \) means \( y \) divides \( x \) and \( 1 < y < x \).
A given formula may have different interpretations under which it is true and others under which it is false.

Metaphorically, one is looking at different worlds. In some, a given statement may be perfectly acceptable. In others, it might be utter nonsense.

In normal life and in the context of programming we are already familiar with this idea of world.

When a program is to find a solution to a certain problem, we are really trying to find a world in which the specification of our problem is true.
Suppose $\Sigma$ is a set of formulas in $\mathcal{L}^{\text{pred}}$. We define

$$[I, s](\Sigma) = \begin{cases} 1 & \text{if for every } B \in \Sigma, [I, s](B) = 1, \\ 0 & \text{otherwise} \end{cases}$$

When $\Sigma \subseteq \text{Sent}(\mathcal{L}^{\text{pred}})$, $s$ does not matter at all.
Definition: Satisfiability

- $\Sigma \subseteq \text{Sent}(\mathcal{L}^{pred})$ is satisfiable iff there is some interpretation $I$ such that $I(\Sigma) = 1$.
- $\Sigma \subseteq \text{Form}(\mathcal{L}^{pred})$ is satisfiable iff there is some interpretation $I$ with some value assignment $s$ such that $[I, s](\Sigma) = 1$.
- When $\Sigma \subseteq \text{Sent}(\mathcal{L}^{pred})$ and $I(\Sigma) = 1$ we say that $I$ satisfies $\Sigma$, or $I$ is a model of $\Sigma$, or $\Sigma$ is true in $I$.
- When $\Sigma \subseteq \text{Form}(\mathcal{L}^{pred})$ and $[I, s](\Sigma) = 1$ we say that $I$ satisfies $\Sigma$ with $s$, or that $s$ satisfies $\Sigma$ within $I$. 
Definition: Validity

• A sentence (closed formula) $A \in \text{Sent}(\mathcal{L}^{\text{pred}})$ is valid iff for every interpretation $I$, $I(A) = 1$.

• A formula $A \in \text{Form}(\mathcal{L}^{\text{pred}})$ is valid iff for every interpretation $I$ and every value assignment $s$ in $I$, $[I, s](A) = 1$.

• Validity is also called universal validity.
• A **valid formula** is one that is true on account of its form alone, irrespective of the meaning of the non-logical symbols (parameters) and the free variable symbols yielded under interpretations and value assignments.

• Validity is intended to capture the informal notion of truth of propositions with attention to the logical form, and in abstraction from the matter.

• A **satisfiable formula** (or set of formulas) is one that is true relative to some particular interpretation and value assignment. Hence, satisfiability corresponds to the informal notion of truth of propositions which follows from the matter.
Example

Suppose $A = f(g(a), g(u)) \approx g(b)$, $I$ is an interpretation over the domain $D = \mathbb{N}$ and $s$ is an assignment in $I$ such that $I(a) = 3$, $I(b) = 5$, $s(u) = 4$, $I(f)(x, y) = x + y$, and $I(g)(x) = x^2$.

Then $[I, s](A)$ is the true proposition

$$3^2 + 4^2 = 5^2.$$ 

Hence, $A$ is **satisfiable**. The truth of (1) is determined by the matter. In fact, there are other interpretations and assignments which make $A$ true.

$A$ is **not valid**. If we set $I(b) = 6$ in the above interpretation, $[I, s](A)$ will be false.
Example

Suppose now that $B = F(u) \lor \neg F(u)$, I and s are any interpretations and assignment respectively.

Then, $[I, s](B)$ is the true proposition

$$(2) \quad s(u) \text{ has or has not the property } I(F).$$

The truth of (2) is not concerned with the domain, the individual $s(u)$ or the property $I(F)$. It follows from the logical form which justifies the validity of $B$. 
Remarks

- Valid formulas in $\mathcal{L}^{pred}$ are the counterpart of tautologies in $\mathcal{L}^p$.
- The similarities between them are obvious, but there is one important difference.
- To decide whether or not a formula of $\mathcal{L}^p$ is a tautology, algorithms are used (for instance the truth table method).
- However, in order to know whether a formula of $\mathcal{L}^{pred}$ is valid, we have to consider all interpretations and all value assignments over domains of different sizes.
• In case of an infinite domain, the procedure is in general not finite.

• Given an interpretation we are not provided a method for evaluating the value of $\forall x B(x)$ or $\exists x B(x)$ in a finite number of steps, because it presupposes the values of $B(u)$ for infinitely many $u$ in $D$.

• It is sometimes possible to decide for certain formulas of $\mathcal{L}^{\text{pred}}$ whether they are valid or not.

• However, in the general case we have the following result.
Theorem There is no algorithm for deciding the validity or satisfiability of formulas in $\mathcal{L}^{\text{pred}}$. (Church, 1936)

Alonzo Church (1903 -1995)
• In **first-order logic**, the variables range over individuals from the domain. The quantifiers are interpreted in the familiar way as “for all individuals of the domain” and “there exist some individual of the domain”.

• In **second-order logic**, we allow as variables subsets of the domain and relations on the domain. For instance, in the following propositions
  - Each non-empty subset of natural numbers has a smallest element
  - $\forall P \forall x (x \in P \lor x \notin P)$
    (for every unary relation (or set) $P$ of individuals, and every individual $x$, either $x$ is in $P$ or it is not)
  we have to take all subsets of the domain into consideration and require variables and quantifiers for sets.

• In **higher-order logic**, variables and quantifiers for sets of sets, sets of sets of sets, etc will be allowed.
History

- Aristotle (384 - 322 BC) - earliest study of formal logic
- George Boole (1815-1864) - propositional logic
- Gottlob Frege (1848-1925) - predicate logic

- Euclid’s Theorem (that there are infinitely many primes) can be expressed in predicate logic but not propositional logic
- Predicate logic was intended to express all mathematics.
Frege’s second-order logic

- Frege did not restrict to first-order logic. He thought he put all mathematics on firm logical foundation. Just before the publication of Frege’s work, Bertrand Russell (1872-1970) pointed out a paradox at the heart of Frege’s system:

\[ R = \text{Set consisting of all sets that are not members of themselves.} \]

- Is \( R \) a member or itself?
Why restrict to first-order logic?

• “Hardly anything more unfortunate can befall a scientific writer than to have one of the foundations of his edifice shaken after the work is finished. This was the position I was placed in by a letter of Mr. Bertrand Russell, just when the printing of this volume was nearing its completion.” (Frege)

• Restricting to first-order logic is one way to avoid Russell’s paradox.