

Undecidability

Several key questions about predicate logic, such as the validity and satisfiability problems, are undecidable.

Validity Problem.

Given a predicate logic formula ϕ , is ϕ valid or not?

An algorithm solving this problem would have to terminate for all input formulas and correctly answer the stated question. No such algorithm exists, as the following, *unsolvable*, problem can be reduced to the validity problem:

Post Correspondence Problem

Given a finite sequence,

$$(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k),$$

of pairs of nonempty binary strings, is there a sequence of indices i_1, \dots, i_n with $n \geq 1$, such that the concatenated strings $s_{i_1}s_{i_2} \cdots s_{i_n}$ and $t_{i_1}t_{i_2} \cdots t_{i_n}$ are identical?

For instance, for the pairs

$$(1, 101), (10, 00), (011, 11),$$

one obtains a solution by choosing the sequence of indices 1, 3, 2, 3.

Reduction

Theorem

The validity problem for predicate logic is undecidable.

Sketch of proof.

We outline how a decision procedure for the validity problem can be used to solve Post's correspondence problem.

Let C be an instance $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$ of the correspondence problem. We effectively construct a formula ϕ that is valid if, and only if, C has a solution.

First we encode bitstrings as terms using a constant e and unary function symbols f_0 and f_1 . Intuitively, e represents the empty string, whereas f_0 and f_1 represent adding one bit, 0 or 1, to the end of string. A bitstring $b_1b_2 \dots b_l$ is encoded by the term

$$f_{b_l}(f_{b_{l-1}}(\dots(f_{b_1}(e))\dots)).$$

We also use the expression $f_{b_1b_2 \dots b_l}(t)$ as an abbreviation for the term $f_{b_l}(f_{b_{l-1}}(\dots(f_{b_1}(t))\dots))$.

For example, the bitstring 011 is encoded as $f_1(f_1(f_0(e)))$.

In addition, let P be a binary predicate symbol and ϕ be the formula $\phi_1 \wedge \phi_2 \rightarrow \phi_3$, where

$$\phi_1: \bigwedge_{i=1}^k P(f_{s_i}(e), f_{t_i}(e))$$

$$\phi_2: \forall v \forall w [P(v, w) \rightarrow \bigwedge_{i=1}^k P(f_{s_i}(v), f_{t_i}(w))]$$

$$\phi_3: \exists z P(z, z)$$

Intuitively, $P(s, t)$ is meant to express that, for some sequence of indices i_1, \dots, i_n , the term s equals $s_{i_1} \cdots s_{i_n}$ and t equals $t_{i_1} \cdots t_{i_n}$.

It can be shown that the correspondence problem C has a solution if, and only if, the corresponding formula ϕ is valid.

Note

The validity problem is *semi-decidable*. That is, one can design a procedure (e.g., one based on Skolemization and resolution) that correctly determines whether a given formula is valid. This procedure need not terminate for input formulas that are not valid.

Compactness

A key result about first-order predicate logic is the following

Compactness Theorem

A set of predicate logic sentences N is satisfiable if and only if every *finite* subset of N is satisfiable.

Proof. (1) If N is satisfiable, then obviously every subset of N is also satisfiable.

(2) Suppose N is not satisfiable. Let N' be a set of clauses obtained from N via Skolemization. Then N' is also unsatisfiable and thus the empty clause can be derived from N' by resolution. Any derivation involves only clauses in a finite subset N'_0 of N' . There is a corresponding finite subset N_0 of N from which all clauses in N'_0 (and possibly other clauses) can be derived. Since N'_0 is unsatisfiable, the set N_0 is also unsatisfiable. ■

Non-standard Models

The Compactness Theorem can be used to prove the existence of non-standard models of first-order arithmetic.

Consider the first-order language with function symbols $+$, \cdot , 0 , and 1 and predicate symbol $<$.

Let \mathbf{N} be a model for this language with the natural numbers as domain and the standard interpretation of the symbols as addition, multiplication, the numbers zero and one, and the less-than relation. In addition, let S be the set of all predicate logic sentences that are true in this model, i.e., $S = \{F : \mathbf{N} \models F\}$.

We can use the compactness theorem to show that there are “non-standard” models of S , different from \mathbf{N} .

Let c be a constant and S_c be the set S plus all sentences $0 < c$, $1 < c$, $1 + 1 < c$, $1 + (1 + 1) < c$, \dots

If S' is a finite subset of S_c , we can extend the structure \mathbf{N} to a model of S' by simply assigning to the constant c a suitable (i.e., large enough) value so that all inequalities $t < c$ in S' are satisfied.

Thus all finite subsets of S_c are satisfiable. Hence by the Compactness Theorem, the set S_c itself is satisfiable. But any model of S_c is also a model of S , different from the standard model \mathbf{N} .

Reachability

The following example also illustrates limitations to the expressiveness of first-order predicate logic.

Reachability Problem

Let $G = (V, E)$ be a directed graph. Given nodes n and n' in V , is there a finite path from n to n' in G ?

The existence of a path of any *fixed* length k between u and v can be expressed in predicate logic, e.g., by the formulas

$$\phi_1 = R(u, v) \text{ and}$$

$$\phi_k = \exists x_1 \cdots \exists x_{k-1} [R(u, x_1) \wedge R(x_1, x_2) \wedge \cdots \wedge R(x_{k-1}, v)]$$

if $k > 1$,

where R represents the set of edges E .

But the existence of *some* path of finite length between u and v can not be expressed in predicate logic.

Suppose ϕ is a formula with free variables u and v , and using only logical symbols and the binary predicate symbol R , that expresses the existence of a path from u to v in the graph represented by R .

Let c and d be constants and Δ be the following set of formulas:

$\neg(c \equiv d),$
 $\neg\phi_i[c/u, d/v], i \geq 1,$ and
 $\phi[c/u, d/v].$

The last formula in Δ asserts that the the nodes represented by c and d are connected by a path, whereas the other formulas express that c and d are different and not connected by any finite path.

Every finite subset of Δ is satisfiable. Thus, by the Compactness Theorem, Δ is satisfiable, which is a contradiction. We conclude that there is no formula that expresses reachability.

Equality

The intended semantics of the equality predicate \equiv is that, for all models \mathcal{M} , $a \equiv^{\mathcal{M}} b$ if and only if a and b are the same elements of the underlying universe A .

Equality can be axiomatized by the following formulas:

$$x \equiv x,$$

$$x_1 \equiv y_1 \wedge \cdots \wedge x_n \equiv y_n \rightarrow f(x_1, \dots, x_n) \equiv f(y_1, \dots, y_n),$$

for all function symbols f (of arity n), and

$$x_1 \equiv y_1 \wedge \cdots \wedge x_n \equiv y_n \wedge P(x_1, \dots, x_n) \rightarrow P(y_1, \dots, y_n),$$

for all predicate symbols P (of arity n).

Second-Order Logic

Reachability can be expressed in *second-order logic*, which allows quantification over predicate variables.

Consider the following formulas:

$$\psi_1: P(x, x)$$

$$\psi_2: P(x, y) \wedge P(y, z) \rightarrow P(x, z)$$

$$\psi_3: \neg P(u, v)$$

$$\psi_4: R(x, y) \rightarrow P(x, y)$$

The first two formulas express that the relation denoted by P is reflexive and transitive. The last formula states that the relation denoted by R is contained in the relation denoted by P .

The second-order formula

$$\exists P \forall x \forall y \forall z [\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4]$$

specifies *unreachability*, whereas its negation

$$\forall P \exists x \exists y \exists z [\neg \psi_1 \vee \neg \psi_2 \vee \neg \psi_3 \vee \neg \psi_4]$$

expresses reachability.

Another Example

Recall that a Horn clause contains at most one positive literal:

$$B \quad (1)$$

$$\neg A_1 \vee \cdots \vee \neg A_k \vee B \quad (2)$$

$$\neg A_1 \vee \cdots \vee \neg A_k \quad (3)$$

where A_1, \dots, A_k, B are atomic formulas.

Let $\sigma_1, \sigma_2, \dots$ be a (possibly infinite) sequence of truth assignments to atomic formulas. Define the *product* σ of the sequence as follows:

$$A\sigma = \begin{cases} 1 & \text{if } A\sigma_i = 1 \text{ for all } i \\ 0 & \text{otherwise} \end{cases}$$

1. If a set of clauses S is true under each assignment σ_i , is S also true under the product σ ?
2. If a set of Horn clauses S is true under each assignment σ_i , is S also true under the product σ ?
3. Let S' be the set of all negative unit clauses $\neg A$, such that A is not a logical consequence of S . Is the set $S \cup S'$ satisfiable, provided S is a satisfiable set of Horn clauses?

An Application of the Infinity Lemma

Let S be an infinite set of numbers, presented in binary notation. There is an infinite sequence of different binary numbers

$$b_1, b_2, \dots$$

such that each bitstring b_i is a prefix of b_{i+1} and also a prefix of some element of S .

Proof.

We may assume, without loss of generality, that S does not contain 0, so that each binary number begins with a 1. Define a labeled binary tree representing all prefixes of binary numbers in S as follows.

Each node is labeled by 0 or 1; the root of the tree, by 1. A node represents the binary number specified by the bits on the path from the root to that node. (Since the root is labeled by 1, only binary numbers beginning with a 1 are represented.) If a node represents a binary number b , it will have a child labeled by 0, if and only if $b0$ is a prefix of some number in S ; and a child labeled by 1, if and only if $b1$ is a prefix of some number in S .

Since S is infinite, the tree representing all prefixes of elements of S must be infinite. It is also

a finitely branching tree, as a node may have at most two children. By the Infinity Lemma, the tree must have an infinite branch. The binary numbers represented by successive nodes along any infinite branch, beginning with the root, form an infinite sequence b_1, b_2, \dots such that each b_i is a prefix of b_{i+1} and also a prefix of some element of S .