$$B_1, \ldots, B_n \rightarrow$$

with the variables $X_1, \ldots, X_k$, we answer positively if, and only if,

$$pl(P) \models \exists X_1 \ldots \exists X_k (B_1 \wedge \ldots \wedge B_n) \tag{1}$$

or equivalently, if

$$pl(P) \cup \{\neg \exists X_1 \ldots \exists X_k (B_1 \wedge \ldots \wedge B_n)\} \text{ is unsatisfiable} \tag{2}$$

In other words, we give a positive answer if the predicate logic representation of the program P, together with the predicate logic interpretation of the query, is unsatisfiable (a contradiction).

The formal definition of the semantic concepts of predicate logic is found in the literature. Here we just give an informal presentation. The components of the logical language (signature) may have any meaning we like. A predicate logic model, $\mathcal{A}$, assigns a certain meaning. In particular, it consists of

- a *domain* $dom(\mathcal{A})$, a nonempty set of objects about which the formulas make statements,
- an element from the domain for each constant,
- a concrete function on $dom(\mathcal{A})$ for every function symbol,
- a concrete relation on $dom(\mathcal{A})$ for every predicate.

When the symbol $=$ is used to denote equality (i.e., its interpretation is fixed), we talk of *Horn logic with equality*. The meanings of the logical connectives $\neg, \vee, \wedge, \rightarrow, \forall, \exists$ are defined according to their intuitive meaning: not, or, and, implies, for all, there is. This way we define when a formula is true in a model $\mathcal{A}$, denoted as $\mathcal{A} \models \varphi$.

A formula φ *follows* from a set M of formulas if φ is true in all models $\mathcal{A}$ in which M is true (that is, all formulas in M are true in $\mathcal{A}$).