Philosophy of Logic – Reexamining the Formalized Notion of Truth

Because formal systems of symbolic logic inherently express and represent the deductive inference model formal proofs to theorem consequences can be understood to represent sound deductive inference to deductive conclusions without any need for other representations.

To put this in laymen's terms all of the truth that can be expressed using words or math symbols is anchored in sentences that are defined to be true: “A cat is an animal”.
Other true sentences are derived from this basic set:
(1) A cat is an animal.
(2) Animals breath.
(3) Therefore cats breath.

The basic truths of English would be called axioms in math.
The derived truths of English would be called theorems in math.
It turns out that all conceptual truth works this same way.

I am approaching this analysis from the frame of reference of the Tarski Undefinability Proof. Minimal Type Theory was created as a universal Tarski metalanguage eliminating the need to switch back and forth and mix and match between a meta-language and a separate object language. MTT is its own meta-language, can express any level of logic and has its own provability operator: “⊢”. (see appendix for formal specification of Minimal Type Theory)

Instead of Tarski's unnecessarily convoluted analysis:

> Since, moreover, the metatheory can be interpreted in the theory enriched by variables of higher order (cf. p. 184) and since in this interpretation the sentence x, which contains no specific term of the metatheory, is its own correlate, the proof of the sentence x given in the metatheory can automatically be carried over into the theory itself: the sentence x which is undecidable in the original theory becomes a decidable sentence in the enriched theory.

We refer to this Tarski definition:

> the metalanguage to be so constructed that the language we are studying forms a fragment of it ; every expression of the language is at the same time an expression of the metalanguage,

https://en.wikipedia.org/wiki/Theory_(mathematical_logic)
The construction of a theory begins by specifying a definite non-empty conceptual class E the elements of which are called statements. These initial statements are often called the primitive elements or elementary statements of the theory, to distinguish them from other statements which may be derived from them.

A theory T is a conceptual class consisting of certain of these elementary statements. The elementary statements which belong to T are called the elementary theorems of T and said to be true. In this way, a theory is a way of designating a subset of E which consists entirely of true statements. (Curry 2010).

From this basis we can infer that every formal proof of theorems in such a (Curry 2010) formal system would exactly correspond to deriving the conclusion of sound deductive inference. (Braithwaite 1962: 2) explains this in depth below:

In order to show that in a deductive system every theorem follows from the axioms according to the rules of inference it is necessary to consider the formulae which are used to express the axioms and theorems of the system, and to represent the rules of inference by rules Gödel calls them "mechanical" rules, p. 37) according to which from one or more formulae another formula may be obtained by a manipulation of symbols. Such a representation of a deductive system will consist of a sequence of formulae (a calculus) in which the initial formulae express the axioms of the deductive system and each of the other formulae, which express the theorems, are obtained from the initial formulae by a chain of symbolic manipulations. **The chain of symbolic manipulations in the calculus corresponds to and represents the chain of deductions in the deductive system.** **(**Braithwaite 1962: 2**)**

But this correspondence between calculus and deductive system may be viewed in reverse, and by looking at it the other way round Hilbert originated metamathematics. Here a calculus is constructed, independently of any interpretation.

From the above we can see that the formal proof to theorem consequences expressed in symbolic logic represents and expresses sound deductive inference to deductive conclusions. One way to look as this might be that formal proof to theorem consequences corresponds to and expresses the sound deductive inference model.

Within the (Braithwaite 1962: 2) correspondence between formal proof and deductive inference it is impossible to have any sound deduction that is not also a formal proof to a theorem consequence.

Within the (Curry 2010) definition of formal system the semantic truth value of axioms is propagated to theorem consequences (because valid deduction is truth preserving) without the need for any alternative system of representation such as model theory. These two views taken together provide the basis for this universal Truth predicate:

$\forall F \in Formal\_Systems\ \forall x \in WFF(F)\ (True(F, x) \leftrightarrow (F \vdash x))$

Thus showing that truth cannot possibly diverge from provability, within this (Braithwaite / Curry) analytical framework. Thus the following sentence would be **unsatisfiable within this framework:** $\exists F \in Formal\_Systems\ (\exists G \in Language(F)\ (G \leftrightarrow \sim(F \vdash G)))$

$G \leftrightarrow \sim(F \vdash G)$ This means that G always has the same Truth value as its unprovability.
(a) if G is unprovable in F then G would be True in F.
(b) if G is provable in F then G would be False in F.

Within this (Braithwaite / Curry) analytical framework G is only True in F when G is a theorem of F thus provable in F. This makes: $G \leftrightarrow \sim(F \vdash G)$ self-contradictory. G must be provably unprovable in F. Self-contradictory expressions of language are treated in the deductive inference model as if they has contradictory premises, thus unsound.

References

Braithwaite, R. B. 1962. On Formally Undecidable Propositions of Principia Mathematica And Related Systems Introduction by R. B. Braithwaite.

Curry, Haskell. 2010 Foundations of Mathematical Logic.

MTT is intended to be used as a universal Tarski meta-language including a meta-language to itself. Because MTT has its own provability operator: “⊢” provability can be directly analyzed directly within the deductive inference model instead indirectly through diagonalization. This allows us to see exactly why an expression of language can be neither proved nor disproved, details that diagonalization cannot provide.

```
%left  IDENTIFIER          //  Letter+ (Letter | Digit)*  // Letter includes UTF-8
%left  SUBSET_OF           //  ⊆
%left  ELEMENT_OF          //  ∈
%left  FOR_ALL             //  ∀
%left  THERE_EXISTS        //  ∃
%left  IMPLIES             //  →
%left  PROVES              //  ⊢
%left  IFF                 //  ↔
%left  AND                 //  ∧
%left  OR                  //  ∨
%left  NOT                 //  ~
%left  ASSIGN_ALIAS        //  :=  LHS is assigned as an alias name for the RHS (macro substitution)
%%

sentence
      : atomic_sentence
      | '~' sentence %prec NOT
      | '(' sentence ')'
      | sentence   IMPLIES      sentence
      | sentence   IFF          sentence
      | sentence   AND          sentence
      | sentence   OR           sentence
      | quantifier IDENTIFIER   sentence
      | quantifier IDENTIFIER   type_of IDENTIFIER sentence  // Enhancement to FOL
      | sentence   PROVES       sentence                     // Enhancement to FOL
      | IDENTIFIER ASSIGN_ALIAS sentence                     // Enhancement to FOL
      ;

atomic_sentence
      : IDENTIFIER '(' term_list ')' // ATOMIC PREDICATE
      | IDENTIFIER                   // SENTENTIAL VARIABLE  // Enhancement to FOL
      ;

term
      : IDENTIFIER '(' term_list ')' // FUNCTION
      | IDENTIFIER                   // CONSTANT or VARIABLE
      ;

term_list
      : term_list ',' term
      | term
      ;

type_of
     : ELEMENT_OF                                           // Enhancement to FOL
     | SUBSET_OF                                            // Enhancement to FOL
     ;

quantifier
     : THERE_EXISTS
     | FOR_ALL
     ;
```