The Notion of Truth in Natural and Formal Languages

For any natural (human) or formal (mathematical) language L we know that an expression X of language L is true if and only if there are expressions Γ of language L that connect X to known facts.

By extending the notion of a Well Formed Formula to include syntactically formalized rules for rejecting semantically incorrect expressions we recognize and reject expressions that evaluate to neither True nor False.

The foundation of this system requires the notion of a BaseFact that anchors the semantic notions of True and False. When-so-ever a formal proof from BaseFacts of language L to a closed WFF X or ~X of language L does not exist X is decided to be semantically incorrect.

A language L is a set of finite strings of characters from a defined alphabet specifying relations to other finite strings. These finite strings could be tokenized as single integer values.

An Minimal Type Theory Relation is the same as a Predicate from Predicate Logic, essentially a Boolean valued function that takes one of more arguments. Minimal Type Theory syntax is nearly identical to First Order Predicate Logic except that it coalesces every finite order of logic from Zero to N into a single simple language.

An Axiom / Postulate is a proposition regarded as self-evidently true without proof. The concept of a BaseFact unifies the mathematical concept of an Axiom / Postulate across natural and formal language. It also formalizes how Axioms / Postulates are defined within language.

A BaseFact is an expression X of (formal or formalized natural) language L that has been assigned the semantic property of True by making it a member of the collection named: BaseFacts.
(1) BaseFacts that contradict other BaseFacts are prohibited.
(2) BaseFacts must specify Relations between Things.
Finite string Expression X expresses relation R of language L.

The above is the complete specification for a BaseFact.

To verify that an expression X of language L is True or False only requires a syntactic logical consequence inference chain (formal proof) from one or more BaseFacts to X or ~X. (Backward chaining reverses this order).

$\text{True}(L, X) \equiv \exists \boldsymbol{\Gamma} \subseteq \text{BaseFacts}(L)\ (\boldsymbol{\Gamma} \vdash X)$
$\text{False}(L, X) \equiv \exists \boldsymbol{\Gamma} \subseteq \text{BaseFacts}(L)\ (\boldsymbol{\Gamma} \vdash {\sim}X)$

Sentence (mathematical logic)
In mathematical logic, a sentence of a predicate logic is a boolean-valued well-formed formula with no free variables. A sentence can be viewed as expressing a proposition, something that must be true or false. The restriction of having no free variables is needed to make sure that sentences can have concrete, fixed truth values: As the free variables of a (general) formula can range over several values, the truth value of such a formula may vary.

Defining a Generic Decidability Decider:
$\forall L \in \text{Formal\_Systems}$
$\forall X \in \text{Closed-WFF}(L)$
$\sim\text{True}(L, X) \land \sim\text{False}(L, X) \rightarrow \text{Incorrect}(L, X)$

The satisfaction of the above formulas indicates Truth or Falsehood.
$\text{True}(L, X) \equiv \exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash X)$
$X \equiv \exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash X)$

$\exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash [\ X \equiv \exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash X)\ ])$
It does not seem to make a difference when we plug the body of X into the body of True(L, X). We are at most evaluating X at two distinct logic levels: Whether or not there are any BaseFacts showing that there are BaseFacts, might be compressed into a single logic level.

Now that we have a pair of formulas defining the notion of True and False we can reevaluate paradoxical expressions differently. The Truth Teller Paradox and one variation of the Liar Paradox simply become false.

Pathological self-reference retains its original definition: Any expression of language that cannot be evaluated to True or False because of self-reference has the semantic error of Pathological self-reference.

"This sentence is True." // Truth Teller Paradox is FALSE
$X \equiv \exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash X)$
"There exists a BaseFact antecedent deriving this sentence as the consequent."

"This sentence is False." // Liar Paradox is FALSE
$Y \equiv \exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash \sim Y)$
"There exists a BaseFact antecedent deriving the negation of this sentence as the consequent."

"This sentence is logically equivalent to this sentence is True."
$LP \leftrightarrow \exists\Gamma \subseteq \text{BaseFacts}(L)\ (\Gamma \vdash LP)$ // Truth Teller Paradox is FALSE
[This sentence is logically equivalent to]
[There exists a BaseFact antecedent deriving this sentence as the consequent]

“This sentence asserts that five is greater than three”
Y ≡ Assert_Relation(Y, “5 > 3”)   // Non paradoxical is TRUE
[01] Assert_Relation (1)(2)
[02] > (3)(4)
[03] “5”
[04] “3”

“This sentence is Not True.”   // Liar Paradox is neither True nor False
$Z \equiv \sim\exists\Gamma \subseteq BaseFacts(L)\ (\Gamma \vdash Z)$
“There does not exist a BaseFact antecedent deriving this sentence as the consequent.”

“This sentence is Not Provable”   // 1931 GIT is neither True nor False
$G \equiv \sim\exists\Gamma \subseteq WFF(L)\ (\Gamma \vdash G)$
“There does not exist a WFF antecedent deriving this sentence as the consequent.”

These last two sentences both have True assertions that do not make their sentences True because that would contradict their assertion making it False. These sentences are also not False because their assertion is True.