

# Axiomatic Extensions of Turing Machines are Equivalent to Oracles

Valerii Kremnev

January 2026

“Let us suppose we are supplied with some unspecified means of solving number-theoretic problems; a kind of oracle as it were. We shall not go any further into the nature of the oracle apart from saying it cannot be a machine.”

- A. M. Turing, 1939 [2]

## Abstract

Computer algebra systems are implemented as Turing machines, yet they allow one to manipulate symbols representing real numbers and transcendental functions. Richardson [1] established that for expression classes containing elementary functions ( $\exp$ ,  $\sin$ ,  $\pi$ ,  $\log 2$ ), the identity problem - deciding whether an expression equals zero - is generally undecidable. But current literature mostly treats the expressions about reals provided to the TM as given. This raises a question: what is the formal status of axioms that CAS systems use?

We draw a distinction between rules and axioms *accessible* to a Turing machine (decidable through mechanical symbol replacement) and axioms that must be *provided* externally (undecidable, yet accepted in the formal system). We show that providing an inaccessible axiom or rule is functionally isomorphic to providing an oracle in Turing’s sense.

We would like to highlight the applicability of those results to any arbitrary TM which is supplied by externally provided rules or axioms - including modern AI systems. We find the question of the epistemic stance of AI towards the rules discoverable in training data directly relevant for understanding abilities and limitations of such systems.

## Contents

Abstract . . . . .	1
1. Oracle Structure . . . . .	2
2. Rewrite Rule Structure for CAS . . . . .	3
3. Accessible and Inaccessible Rules . . . . .	4
4. Equivalence . . . . .	5
5. Example: $\sin^2(x) + \cos^2(x) = 1$ . . . . .	7
6. CAS Behavior . . . . .	8
7. Conclusion . . . . .	9
References . . . . .	10

## 1. Oracle Structure

**Definition 1** (Oracle Turing Machine). An Oracle Turing Machine for set  $O$  is a Turing machine augmented with:

- An oracle tape
- A query state  $q_?$
- Response states  $q_{\text{yes}}, q_{\text{no}}$

In Turing's original formulation, the machine enters internal configuration  $\mathbf{o}$  with a well-formed formula  $\mathbf{A}$  on the tape, and transitions to configuration  $\mathbf{p}$  or  $\mathbf{t}$  depending on whether  $\mathbf{A}$  is true. We represent  $\mathbf{A}$  as the query string  $w$  throughout.

On entering  $q_?$  with string  $w$  on the oracle tape:

if  $w \in O$  : transition to  $q_{\text{yes}}$ ,    else: transition to  $q_{\text{no}}$

This transition is instantaneous and requires no computation.

### Properties:

Property	Description
Instantaneous	One step, no computation
Unjustified	No proof, just yes/no
Infinite scope	Decides membership in potentially infinite sets
External	Truth imported from outside the system

## 2. Rewrite Rule Structure for CAS

A CAS operates by applying rewrite rules to expressions. We would like to formalize it to show the equivalence to an oracle.

**Definition 2** (Rewrite Rule). A rewrite rule is a pair  $(L, R)$  of expression patterns, written  $L \rightarrow R$ , where  $L$  and  $R$  contain pattern variables  $x_1, \dots, x_n$ .

**Definition 3** (Rule Application). Given expression  $E$  and rule  $L \rightarrow R$ :

1. **Pattern match:** Find substitution  $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$  such that  $L[\sigma] = E$
2. **Rewrite:** Replace  $E$  with  $R[\sigma]$

Pattern matching and substitution are computable and represent pure symbol manipulation.

**Definition 4** (Rule Validity). A rule  $L \rightarrow R$  is *valid* if for all substitutions  $\sigma$ :

$$L[\sigma] = R[\sigma]$$

This is a statement about equality in the underlying mathematical domain.

**Example:** The rule  $\sin^2(x) + \cos^2(x) \rightarrow 1$

Step	Operation	Computation
Match	Find $\sigma$ such that $\sin^2(x) + \cos^2(x)$ matches $E$	Computable
Valid?	Is $\sin^2(t) + \cos^2(t) = 1$ for the matched term $t$ ?	Not computable in general
Rewrite	Replace $E$ with 1	Computable

The CAS performs steps 1 and 3. Step 2 - the validity of the rule - is given. The decision that the rule exists and is correct was provided to the CAS.

**The Oracle Perspective:** A rewrite rule  $L \rightarrow R$  encodes answers to queries about the set of valid equalities:

$$O = \{(E, E') \mid E = E' \text{ is a valid equality}\}$$

The rule pre-answers infinitely many membership queries:

$$\forall \sigma : (L[\sigma], R[\sigma]) \in O \rightarrow \text{Yes|No}$$

When the CAS pattern-matches  $E$  against  $L$  and obtains  $\sigma$ , it queries: is  $(E, R[\sigma]) \in O$ ? The rule provides the answer: “yes” - for existing rule, or “no” - for any other arbitrary input. But does this mean that any rule acts as an oracle and can solve an undecidable problem?

### 3. Accessible and Inaccessible Rules

**Definition 5** (Accessible Rule). A rewrite rule  $L \rightarrow R$  is *accessible* to a TM if the validity of  $L[\sigma] = R[\sigma]$  is decidable for all  $\sigma$ .

**Example:** The rule  $x + 0 \rightarrow x$

Validity follows from ring axioms. Given any term  $t$ , the equality  $t + 0 = t$  can be verified by finite symbolic manipulation within decidable arithmetic.

**Definition 6** (Inaccessible Rule). A rewrite rule  $L \rightarrow R$  is *inaccessible* if no algorithm can decide  $L[\sigma] = R[\sigma]$  for arbitrary  $\sigma$ , or such computation never halts.

**Example:** The rule  $\sin^2(x) + \cos^2(x) \rightarrow 1$

Real numbers are equivalence classes of Cauchy sequences. Determining whether  $\sin^2(\alpha) + \cos^2(\alpha) = 1$  for arbitrary real  $\alpha$  requires:

- Computing  $\sin(\alpha)$  and  $\cos(\alpha)$  for a Cauchy sequence  $\alpha$
- Verifying the sum equals 1

This is undecidable in general. The rule's validity depends on the completeness of  $\mathbb{R}$  and properties of transcendental functions - axioms a TM cannot verify.

The proof of undecidability trivially comes from the fact that the direct computation of the next digit of  $\sin^2(\alpha) + \cos^2(\alpha)$  and subsequent check of equivalence to 1 is not guaranteed to halt at any step of the computation due to continuity of reals. Similarly, stopping the algorithm at arbitrary precision wouldn't guarantee that the current result is equal to 1. This result is well known and documented, though the fact about incomputability of real numbers in finite steps is so trivial that it's not even attributable to any particular citation, and proven through reduction to halting.

## 4. Equivalence

**Theorem 1.** Let  $L \rightarrow R$  be an inaccessible rewrite rule. Then providing this rule to a TM is isomorphic to providing an oracle for the set:

$$O_{L \rightarrow R} = \{(E, E') \mid E \text{ matches } L[\sigma] \text{ and } E' = R[\sigma] \text{ for some } \sigma\}$$

*Proof.* We construct explicit mappings between oracle queries and rule applications.

**The equality oracle.** Define the set of valid equalities:

$$O = \{(E, E') \mid E = E'\}$$

### Forward mapping $\varphi$ : Oracle query $\rightarrow$ Rule application

An oracle query asks:  $(E, E') \in O$ ?

Given rule  $L \rightarrow R$  and expression  $E$ :

1. Pattern match: find  $\sigma$  such that  $L[\sigma] = E$  (computable)
2. Compute  $E' = R[\sigma]$  (computable)
3. Query:  $(E, E') \in O$ ?
4. If yes: apply  $L \rightarrow R$ ; continue
5. If no:  $L$ ; continue

$$\varphi : \text{query}((L[\sigma], R[\sigma]) \in O?) \mapsto \text{apply}(L \rightarrow R, \sigma)$$

### Reverse mapping $\psi$ : Rule application $\rightarrow$ Oracle query

A rule application takes  $E$ , finds  $\sigma$ , returns  $R[\sigma]$ .

This is equivalent to:

1. Form the pair  $(E, R[\sigma])$
2. Assert  $(E, R[\sigma]) \in O$

$$\psi : \text{apply}(L \rightarrow R, \sigma) \mapsto \text{query}((L[\sigma], R[\sigma]) \in O?)$$

### Bijectivity:

- $\varphi \circ \psi = \text{id}$ : Apply rule, form query, apply rule  $\rightarrow$  same result
- $\psi \circ \varphi = \text{id}$ : Form query, apply rule, form query  $\rightarrow$  same query

### Decision-preservation:

Oracle	Rule
$(E, E') \in O?$	Does $L$ match $E$ ?
$q_{\text{yes}}$ : pair is valid	Match: rewrite $E \rightarrow R[\sigma]$
$q_{\text{no}}$ : pair not covered	No match: try next rule or keep

The rule encodes: “for all  $\sigma$ , answer Yes to query  $(L[\sigma], R[\sigma]) \in O?$ ”

### Computational equivalence:

For inaccessible rules:

- Oracle: TM cannot compute membership in  $O$
- Rule validity: TM cannot decide  $L[\sigma] = R[\sigma]$

The rule provides oracle access without oracle computation.

**Pseudocode Example.** If we express this equivalence as pseudocode, we will see that these two programs are structurally identical:

```
# Program 1: Rule-based
def apply_rule(expr: str, pattern: str, replacement: str) -> str:
    sigma = match(pattern, expr)
    if sigma:
        return substitute(replacement, sigma)
    return None

def execute(expr: str, rules: list) -> str:
    for (pattern, replacement) in rules:
        result = apply_rule(expr, pattern, replacement)
        if result:
            return result
    return expr

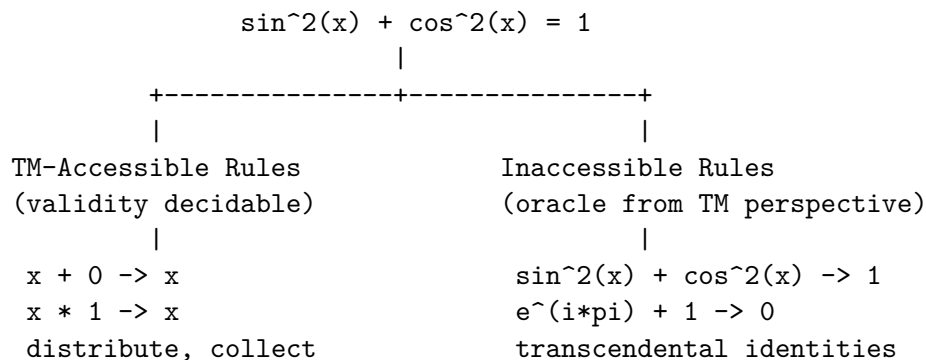
# Program 2: Oracle-based
def check_oracle_set(pair: tuple) -> bool:
    oracle = {} # Can't be constructed from within TM
    return pair in oracle

def execute(expr: str) -> str:
    for candidate in possible_results(expr):
        if check_oracle_set((expr, candidate)):
            return candidate
    return expr
```

As you can see, while in the first section we are “applying the rules”, which looks like a typical TM operation, we could’ve equivalently expressed it as pre-constructing the possible set of checks of pairs. In this case the oracle structure is very clear: we get a separate program which has a set the TM couldn’t express. If we receive yes - we continue execution with rewrite. If we receive no - the expression stays the same, and we either take new rule, or stop attempting to change the expression.

## 5. Example: $\sin^2(x) + \cos^2(x) = 1$

The identity  $\sin^2(x) + \cos^2(x) = 1$  depends on rules from two distinct sources:



The left branch contains rules whose validity a TM can verify: algebraic identities following from ring and field axioms.

The right branch contains rules whose validity a TM cannot verify: they depend on real number equality (undecidable), completeness (non-constructive), and transcendental function properties.

A CAS returns 1 for this query by applying the rule  $\sin^2(x) + \cos^2(x) \rightarrow 1$ :

1. Input:  $\sin^2(x) + \cos^2(x)$
2. Pattern match:  $\sigma = \{x \mapsto x\}$
3. Query (implicit):  $(\sin^2(x) + \cos^2(x), 1) \in O?$
4. Rule answers: Yes
5. Output: 1

The CAS can continue the computation and apply the rewrite only given the oracle output.

## 6. CAS Behavior

The distinction manifests in CAS queries:

```
(* Accessible: algebraic identity, validity decidable *)  
Simplify[Sqrt[2] + Sqrt[3] == Sqrt[5 + 2*Sqrt[6]]]  
(* Output: True *)
```

```
(* Inaccessible: pre-computed oracle output, cached as rule *)  
Simplify[Sin[x]^2 + Cos[x]^2]  
(* Output: 1 *)
```

```
(* No rule available: oracle not consulted for this query *)  
Simplify[E + Pi == SomeExpression]  
(* Output: E + Pi == SomeExpression (unevaluated) *)
```

The first query reduces via TM-accessible rules. The second applies an inaccessible rule (cached oracle output). The third finds no matching rule - the oracle was never queried for this input, so no cached answer exists.

When a CAS returns “unevaluated,” it has not encountered an inaccessible axiom. The query is well-formed, but there is no external truth to continue.



## 7. Conclusion

We have shown that oracles, as defined in Turing's original paper [2], and inaccessible rewrite rules are equivalent.

From the perspective of a Turing machine, an inaccessible rule is a true oracle in exactly Turing's original sense: an external source of truth that gives a yes-or-no answer undecidable from within the TM system.

The implication for AI systems, including the ones with access to tool calling, is that even though they can extract, memoise and use the rules derived from training data, there will be no configuration in which they could verify the validity of those rules and their consistency with each other. From the perspective of an AI system, it's an oracle call, and thus can't be decided from within.

We use “oracle” when we wish to emphasize that the answer comes from beyond computation. We use “axioms” when we have agreed, as a community, that we don’t question a specific rule, accept it as truth, and halt on question where it came from.

Which poses a question: If reality were Turing machines all the way down, where is the source of inaccessible rules?

## References

- [1] D. Richardson, “Some Undecidable Problems Involving Elementary Functions of a Real Variable,” *The Journal of Symbolic Logic*, vol. 33, no. 4, pp. 514–520, 1968.
- [2] A. M. Turing, “Systems of Logic Based on Ordinals,” *Proceedings of the London Mathematical Society*, Series 2, vol. 45, pp. 161–228, 1939.