Rule Externalization and the Decision–Verification Shift

YANLIN LI
Independent Researcher

Abstract

We examine the structural role of rule availability in computational decision problems. By modeling rules as explicit parameters, we formalize a distinction between direct decision under a fixed and internalized rule structure and verification when such structure is externalized or left implicit. We show that externalization of rules induces a shift from direct decision procedures to verification relative to auxiliary information, yielding a problem formulation characteristic of nondeterministic polynomial-time verification. No claims are made regarding separations between complexity classes; the analysis is purely structural.

1 Introduction

Decision problems in complexity theory are typically formulated relative to a fixed and shared rule structure. Under this assumption, the computational task is to decide membership of an instance in a language using a procedure that implicitly incorporates those rules.

In practice and in theory, however, rules governing validity or interpretation may be incomplete, implicit, or external to the decision procedure itself. In such cases, correctness is often established not by direct decision but by verification relative to additional information encoding the missing structure.

This paper formalizes this distinction. We show that when rule structure is externalized, decision tasks naturally assume a verification-based form. The resulting formulation aligns with the standard definition of **NP**, not as a complexity separation but as a change of interface.

2 Rule-Parameterized Decision Problems

Let Σ be a finite alphabet.

Definition 1 (Rule Structure)**.** A *rule structure* is represented by a finite binary string $r \in \{0,1\}^*$.

For each fixed rule structure r, let

$$L_r \subseteq \Sigma^*$$

denote a language determined by r.

The associated decision task is:

$$\mathsf{DECIDE}_r(x) = \begin{cases} 1 & \text{if } x \in L_r, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2 (Internalized Decision)**.** The rule structure r is said to be *internalized* if it is fixed and shared by the decision procedure.

The task DECIDE_r admits *direct efficient decision* if there exists a deterministic polynomial-time algorithm A_r such that

$$A_r(x) = \mathsf{DECIDE}_r(x) \quad \text{for all } x \in \Sigma^*.$$

In this setting, the decision procedure operates entirely within the given rule framework.

3 Externalized Rules and Verification Interfaces

We now consider the case in which the rule structure r is not internalized by the decision procedure. Instead of direct decision, correctness relative to r is established through auxiliary information.

Definition 3 (Verification Predicate)**.** A *verification predicate* is a polynomial-time computable function

$$V : \{0,1\}^* \times \Sigma^* \times \{0,1\}^* \to \{0,1\},$$

where $V(r, x, w) = 1$ certifies that $x \in L_r$ relative to auxiliary information w.

Using V, define the verification language

$$L^{\exists} = \{(r, x) \mid \exists w \in \{0,1\}^* \text{ such that } V(r, x, w) = 1\}.$$

Proposition 1. The language $L^{\exists}$ belongs to **NP**.

Proof. Membership of (r, x) in $L^{\exists}$ is witnessed by a certificate w whose validity can be verified in polynomial time by V. This satisfies the defining property of **NP**. □

4 Decision–Verification Interface Shift

The preceding constructions formalize a structural distinction:

Observation 1 (Rule Externalization and Verification Shift)**.** Efficient decision procedures presuppose a fixed and explicit rule structure internal to the problem formulation. When such structure is externalized or left implicit, the task no longer admits direct decision and instead becomes verification relative to auxiliary information encoding the missing rules.

This shift reflects a change in computational *interface* rather than in computational power. The underlying language family remains unchanged, but the mode by which correctness is established transitions from direct decision to verification.

5 Scope and Non-Claims

This work makes no claims regarding separations between complexity classes, including **P** and **NP**. The constructions presented here do not modify standard definitions and do not assert the existence or non-existence of any class inclusions beyond those definitional in nature.
The purpose of this paper is solely to isolate and formalize the structural effect of rule externalization on decision problem formulation.

6 Conclusion

By treating rule structures as explicit parameters, we have shown that the availability of internalized rules is a necessary condition for direct decision interfaces. When such rules are externalized, decision problems naturally assume a verification-based form consistent with nondeterministic polynomial-time verification.
This observation clarifies a structural mechanism underlying the distinction between decision and verification and may inform further boundary-based analyses of computational problem formulations.