On Explicit Reflection in Theorem Proving and Formal Verification

Sergei N. Artemov *

Department of Computer Science
Cornell University,
Ithaca, NY 14853, U.S.A.
artemov@cs.cornell.edu
http://www.cs.cornell.edu/Info/People/artemov

Abstract. We show that the stability requirement for a verification system yields the necessity of some sort of a reflection mechanism. However, the traditional reflection rule based on the Gödel implicit provability predicate leads to a "reflection tower" of theories which cannot be formally verified. We found natural lower and upper bounds on a metatheory capable of establishing stability of a given verification system.
The paper introduces an explicit reflection mechanism which can be verified internally. This circumvents the reflection tower and provides a strict justification for the verification process. On the practical side, the paper gives specific recommendations concerning the verification of inference rules and building a verifiable reflection mechanism for a theorem proving system.

1 Introduction

There is a large variety of theorem provers and proof checkers which can be used for verification (cf. [8], [1], [11]). The mathematical counterparts of those systems range from first order logic (e.g. in **FOL**) and certain fragments of first order arithmetic to higher order logic (**HOL**), the systems with powerful principles sufficient to accommodate most of the classical mathematics (**Mizar**) and most of the computational and constructive tools (**Nuprl**). The underlying logic of such systems can be either classical or intuitionistic. In this paper we assume that

> *The degree of confidence in facts verified by a certain system is not higher than the degree of confidence in the system itself.*

This paradigm yields the necessity to keep an account of the tools used in a given verification process, including the verification system $\mathcal{V}$ itself along with

* The research described in this paper was supported in part by ARO under the MURI program "Integrated Approach to Intelligent Systems", grant DAAH04-96-1-0341, by DARPA under program LPE, project 34145.

an exact description of the set of all metamathematical assumptions $\mathcal{M}$ made in the process of verification. Therefore, the set of beliefs which the verification is based upon should include $\mathcal{V} \cup \mathcal{M}$. Without loss of generality we assume in this paper that a metatheory $\mathcal{M}$ of a given verification system $\mathcal{V}$ contains $\mathcal{V}$, therefore, $\mathcal{V} \cup \mathcal{M} = \mathcal{M}$.

> For example, suppose we want to verify a statement F by means of the first order arithmetic $\mathcal{PA}$ (i.e. $\mathcal{V} = \mathcal{PA}$). One of the possible ways to put this problem on a formal setting is to say that our goal consists in establishing that $\mathcal{PA} \vdash \mathit{Provable}(F)$, where $\mathit{Provable}(F)$ is a formal statement saying that "F is provable by certain formal tools". Suppose that we have established that $\mathcal{ZF} \vdash \mathit{Provable}(F)$, where $\mathcal{ZF}$ is the Zermelo-Frenkel set theory (a much stronger theory than $\mathcal{PA}$). This corresponds to a realistic situation when a verifier uses the power of all of mathematics, not only the elementary methods formalizable in $\mathcal{PA}$. Here is the sketch of the standard metamathematical argument which under certain assumptions about $\mathcal{ZF}$ concludes that in fact $\mathcal{PA} \vdash \mathit{Provable}(F)$: assume that $\mathcal{ZF}$ is ω-consistent (cf. [15],[7],[16]); since $\mathit{Provable}(F)$ is an arithmetical Σ_1 statement, this yields that $\mathit{Provable}(F)$ is true and, by the Σ_1-completeness of $\mathcal{PA}$, $\mathcal{PA} \vdash \mathit{Provable}(F)$. On the one hand, we have succeeded in establishing that $\mathcal{PA} \vdash \mathit{Provable}(F)$. On the other hand, at the metalevel of this argument we have used the power of $\mathcal{ZF}$ and even the assumption of ω-consistency of $\mathcal{ZF}$. A total account of the beliefs involved in this verification process should include this assumption, which, by the way, has never been and could not possibly be proven by any usual consistent mathematical means.

In this paper we will try to demonstrate the following three points:

1. Some form of the reflection rule is a necessary part of an extendable and stable verification system. This will emerge as a natural corollary of the soundness, extensibility, and stability assumptions (cf. [8]) about a verification system. Moreover, even the most basic proof checking scheme when $\mathcal{V}$ verifies a proof of F and then concludes that F itself holds requires reflection.

2. The traditional reflection based on the implicit provability predicate does not provide a satisfactory justification of formal verification. It is well-known that the implicit reflection in a given system $\mathcal{V}$ cannot be verified in $\mathcal{V}$ (cf. [8], [12], [1], [11]). In particular, this means that implicit reflection cannot justify even the basic proof checking by means of $\mathcal{V}$ without imposing additional unverifiable assumptions on $\mathcal{V}$. The present paper demonstrates that an iterative use of reflection leads to the "reflection tower" of implicit reflection rules which is not computably enumerable and cannot be itself verified by any formal tools. If one takes into account these hidden metamathematical costs of implicit reflection, then no verifiable stable systems exist.

3. There is a verifiable reflection mechanism: "explicit reflection" (introduced in the present paper), which provides a foundational justification of the verification process. Explicit reflection requires more information in order to certify the premises of the reflection rule. However, this additional information is usually available in real processes of verification; the old implicit provability model just has not had a tool of its utilization. The explicit reflection circumvents the reflection tower and provides the strict justification of verification. On the practical side, the paper gives specific recommendations concerning the verification of the admissible rules and building a verifiable reflection mechanism for a theorem proving system.

2 Verification Systems

Definition 1. *Under a* **verification system** $\mathcal{V}$ *we will understand a formal theory satisfying the following conditions a) – d):*

a) The underlying logic of $\mathcal{V}$ *is either classical or intuitionistic.*

b) Proofhood in $\mathcal{V}$ *is decidable, therefore theoremhood in* $\mathcal{V}$ *is computably enumerable. Note that by the well-known Craig Theorem the former follows from the latter for an appropriate choice of axiom system.*

c) $\mathcal{V}$ *is strong enough to represent any computable function and decidable relation.*

d) $\mathcal{V}$ *has some sort of a numeration of syntax mechanism in the style of [8], [1]. In particular, there is an injective function rep which maps syntactic objects like terms, formulas, finite sequences of formulas, sequents, finite trees labeled by sequents, derivation trees, etc., into standard ground terms of* $\mathcal{V}$*. The usual notation used in this case is* $\ulcorner s \urcorner = rep(s)$*. The function rep and its inverse are both computable. We assume that* $\mathcal{V}$ *is able to derive formalizations of "usual" combinatory properties of the syntactic objects at a level corresponding to the first order intuitionistic arithmetic* $\mathcal{HA}$*.*

For the sake of notational simplicity we will use the same names for the informal objects (relations, functions, numbers) and for their formal counterparts (formulas, terms, ground terms) whenever unambiguous.

Examples of verification systems: the first order arithmetic $\mathcal{PA}$; the first order intuitionistic arithmetic $\mathcal{HA}$ and its extensions; second order arithmetic; Martin-Löf type theory $\mathcal{ITT}$; formal set theory $\mathcal{ZF}$; etc. Note that all the above conditions on $\mathcal{V}$ have a purely constructive syntactic character. We have assumed neither semantic properties of $\mathcal{V}$ (e.g. soundness with respect to some semantics), nor metamathematical ones (consistency, ω-consistency, etc.).

Definition 2. *For any verification system* $\mathcal{V}$ *there is a provably* Δ_1 *formula* $Proof(x, y)$ *in the language of* $\mathcal{V}$ *(called* **a proof predicate***) which is obtained by a natural formalization of the inductive definition of derivation in* $\mathcal{V}$ *(cf. [9], [8], [1]). In particular,* $Proof(\ulcorner \mathcal{D} \urcorner, \ulcorner \varphi \urcorner)$ *holds iff* $\mathcal{D}$ *is a proof of* φ *in* $\mathcal{V}$*. The Gödel* **provability predicate** $Provable(y)$ *is defined as* $\exists x Proof(x, y)$*. We will use the notation* $\Box\varphi$ *for* $Provable(\ulcorner \varphi \urcorner)$ *and* $[\![p]\!]\varphi$ *for* $Proof(p, \ulcorner \varphi \urcorner)$*. For any finite set of* $\mathcal{V}$*-formulas* Γ *by* $\Box\Gamma$ *we mean the conjunction of* $\Box\psi$*'s for all* $\psi \in \Gamma$*.*

Definition 3. *The consistency formula* $Consis(\mathcal{V})$ *is defined as* $\neg\Box\bot$*, where* $\bot$ *is the standard boolean constant FALSE in* $\mathcal{V}$*.*

The informal meaning of $Consis(\mathcal{V})$ is that there is no a proof of *FALSE* in $\mathcal{V}$: this is one of the equivalent formulations of the consistency assertion of $\mathcal{V}$ in the language of $\mathcal{V}$.

We will refer to the provability predicate $\Box(\cdot)$ as the *implicit provability predicate*. The reason for choosing this name lies in the fact that in the formula $\Box\varphi$ (i.e. $\exists x Proof(x, \ulcorner\varphi\urcorner)$) the proof is represented implicitly by the existential quantifier, which does not provide any specification of this proof.

The implicit provability predicate has been studied extensively since its invention by Gödel in 1930. The milestone results here are the second Gödel incompleteness theorem (cf. [15], [7])), which states that

$$\textit{If } \mathcal{V} \textit{ is consistent, then } \nvdash Consis(\mathcal{V}),$$

and the Löb theorem which says that

$$\mathcal{V} \vdash \Box\varphi \rightarrow \varphi \textit{ implies } \mathcal{V} \vdash \varphi.$$

By the well-known Hilbert-Bernays lemma (cf. [15],[7]),

$$\mathcal{V} \vdash \varphi \quad \textit{implies} \quad \mathcal{V} \vdash \Box\varphi.$$

This lemma can be considered as a justification of the *formalization rule* $\varphi/\Box\varphi$ for $\mathcal{V}$, which states that every proof in $\mathcal{V}$ can be formalized in $\mathcal{V}$. The proof of the formalization rule is purely syntactic and does not involve any extra assumptions about $\mathcal{V}$. Moreover, this rule can be formalized and proven inside $\mathcal{V}$ (cf. [15], [7]):

$$\mathcal{V} \vdash \Box\varphi \rightarrow \Box\Box\varphi.$$

Below we will use one more fact about the provability operator $\Box$, usually attributed to Hilbert, Bernays and Löb (cf.[15],[7]):

$$\mathcal{V} \vdash \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi).$$

3 Stability Equals Admissibility of Reflection Rule

The basic properties required from a verification system are soundness, extensibility, and stability ([8]). We will discuss soundness in Section 4. Extensibility and stability will appear in this section below.

Definition 4. *A* **rule of inference** *R in the language of $\mathcal{V}$ is a computable partial function from a decidable set of lists of $\mathcal{V}$-formulas to the set of $\mathcal{V}$-formulas.*

The usual notation for a rule of inference R is Γ/φ, where Γ indicates the argument of R (premises), and φ the value $R(\Gamma)$ of R (conclusion). For the sake of notational convenience we will not distinguish between a finite list of formulas Γ and one formula which is the conjunction of all formulas from Γ when unambiguous. We would like to think that such an abuse of notation will be tolerated by a reader.

Definition 5. *A rule of inference Γ/φ in $\mathcal{V}$ is* **derivable** *if for any instance of Γ/φ there is a deduction in $\mathcal{V}$ of its conclusion from its premises.*

A rule of inference Γ/φ in $\mathcal{V}$ is **provable** *if $\mathcal{V} \vdash \Gamma \rightarrow \varphi$.*

A rule of inference Γ/φ in $\mathcal{V}$ is **admissible** *if $\mathcal{V} \vdash \Gamma$ implies $\mathcal{V} \vdash \varphi$.*

Derivable rules is a system with the deduction theorem are provable. It is also natural to assume that every provable rule is derivable since there is a standard deduction of φ from Γ (free premises) and $\Gamma \rightarrow \varphi$ (has a derivation without premises in $\mathcal{V}$.). To a certain extend the notion of provable rule is a system-independent version of the notion of derivable rule.

It is immediate from the definitions that every provable rule of inference is admissible. None of the converse implications holds in a general case. It is important to notice that there are serious reasons to assume that derivable and provable rules alone are not sufficient for an advanced system of automated deduction and verification (cf. also [14]).

Example 1. Here are examples of widely used admissible rules that are not provable: $\varphi/\forall x\varphi$ (*generalization*), $\varphi(x)/\varphi(y)$ (*renaming of free variables*) $\neg\neg\sigma/\sigma$, where σ is a Σ_1 sentence (*Markov rule* for intuitionistic arithmetic $\mathcal{HA}$, cf. [17]), *disjunctive* and *existential* rules for intuitionistic systems, $\varphi/\Box\varphi$ (*formalization*), $\Box\varphi \rightarrow \varphi/\varphi$ (*Löb rule*), $\Box\varphi/\varphi$ (*implicit reflection*, for classical and intuitionistic arithmetic), etc.

Extending $\mathcal{V}$ by adding rules provable in $\mathcal{V}$ does not change the system. In turn, stability of $\mathcal{V}$ with respect to admissible rules rises to a serious theoretical and practical problem since some of the admissible rules cannot be verified inside $\mathcal{V}$.

Definition 6. *A rule of inference Γ/φ in $\mathcal{V}$ is* **verifiable in** $\mathcal{M}$ *if $\mathcal{M} \vdash \Box\Gamma \rightarrow \Box\varphi$. A rule in $\mathcal{V}$ is* **internally verifiable** *if it is verifiable in $\mathcal{V}$.*

Every admissible rule in $\mathcal{V}$ is verifiable in some sufficiently large theory $\mathcal{M}$, but not every one of them is verifiable in $\mathcal{V}$.

Example 2. The rules *generalization, renaming of free variables, formalization, Löb rule* and internally verifiable. The rules *implicit reflection, Markov rule, disjunctive rule, existential rule* are not internally verifiable.

We accept the understanding of stability as conservativity of extensions by internally verifiable rules (cf. [1], [8], [11], [14]).

Definition 7. *System $\mathcal{V}' \supseteq \mathcal{V}$ is* **conservative** *over $\mathcal{V}$ if for any formula ψ*

$$\mathcal{V}' \vdash \psi \quad \textit{implies} \quad \mathcal{V} \vdash \psi.$$

A system $\mathcal{V}$ is **stable** *if for any rule Γ/φ verifiable in $\mathcal{V}$ the system $\mathcal{V} + \Gamma/\varphi$ is conservative over $\mathcal{V}$.*

Definition 8. *By IRR($\mathcal{V}$) we denote the implicit reflection rule $\Box\varphi/\varphi$ where $\Box\varphi$ represents the provability of φ in $\mathcal{V}$.*

Example 3. Here is the standard example of a formal theory for which the implicit reflection rule is not admissible ([9]): $\mathcal{V} = \mathcal{PA} + \neg Consis(\mathcal{PA})$. This system is consistent, i.e. $\mathcal{V} \nvdash \bot$. On the other hand $\mathcal{V} \vdash \Box\bot$, where $\Box$ stands for provability in this particular $\mathcal{V}$.

Theorem 1. *A verification system $\mathcal{V}$ is stable iff the implicit reflection rule $IRR(\mathcal{V})$ is admissible in $\mathcal{V}$.*

Proof. Let $\mathcal{V}$ be a stable system. Let us consider the rule R_φ consisting of a single pair $(TRUE, \varphi)$, where $TRUE$ is the propositional constant for true statements in $\mathcal{V}$. Since $\mathcal{V} \vdash TRUE$, we also have $\mathcal{V} \vdash \Box(TRUE)$. By stability of $\mathcal{V}$, for all φ, ψ if $\mathcal{V} \vdash \Box(TRUE) \to \Box\varphi$ and $\mathcal{V} + TRUE/\varphi \vdash \psi$, then $\mathcal{V} \vdash \psi$. Equivalently, for all φ, ψ if $\mathcal{V} \vdash \Box\varphi$ and $\mathcal{V} + \varphi \vdash \psi$, then $\mathcal{V} \vdash \psi$. Let ψ be φ. Then $\mathcal{V} \vdash \Box\varphi$ implies $\mathcal{V} \vdash \psi$ for all φ, therefore $IRR(\mathcal{V})$ is admissible in $\mathcal{V}$.

Let now $IRR(\mathcal{V})$ be admissible in $\mathcal{V}$, i.e. $\mathcal{V} \vdash \Box\varphi$ implies $\mathcal{V} \vdash \varphi$, and let Γ/φ be a verified rule, i.e. $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$. By an induction on the derivation in $\mathcal{V} + \Gamma/\varphi$ we prove that $\mathcal{V} + \Gamma/\varphi \vdash \psi$ implies $\mathcal{V} \vdash \psi$. The induction basis holds because $\mathcal{V}$ and $\mathcal{V} + \Gamma/\varphi$ have the same set of axioms. The induction step in the case of a rule other than Γ/φ is trivial. Let ψ be obtained in $\mathcal{V} + \Gamma/\varphi$ by the rule Γ/φ, i.e. there is specific Γ_1 such that Γ_1/ψ is a special case of the rule Γ/φ and $\mathcal{V} + \Gamma/\varphi \vdash \Gamma_1$. By the induction hypothesis, $\mathcal{V} \vdash \Gamma_1$. By the formalization rule in $\mathcal{V}$, $\mathcal{V} \vdash \Box\Gamma_1$. Since the rule Γ/φ is verified, we have $\mathcal{V} \vdash \Box\Gamma_1 \to \Box\psi$, therefore $\mathcal{V} \vdash \Box\psi$. By the rule $IRR(\mathcal{V})$, $\mathcal{V} \vdash \psi$. ∎

Note 1. Such systems as **LCF, Nuprl, HOL** have the extension mechanisms of *tactics* based on the representation of provable rules. This use of tactics can be justified inside the system and the stability with respect to tactics can be established without any additional assumptions. A general case of stability with respect to all verified rules (including the ones that are not provable) was considered in [1], [8], [11], [14]. It follows from Theorem 1 that the general stability of a given verification system $\mathcal{V}$ is equivalent to admissibility of the implicit reflection rule in $\mathcal{V}$.

4 Provability Tests and other Sources of Reflection

Stability of a verification system with respect to verifiable rules is not the only place where reflection rule becomes necessary.

The basic proof checking scheme when $\mathcal{V}$ verifies a proof of φ in $\mathcal{V}_1$ and then concludes that $\mathcal{V}_1 \vdash \varphi$ requires some sort of reflection too. In the terms of implicit provability this proof checking scheme is the rule

$$\mathcal{V} \vdash \Box_1\varphi \;\Rightarrow\; \mathcal{V}_1 \vdash \varphi,$$

where $\Box_1$ stands for the provability predicate in $\mathcal{V}_1$. In particular, when $\mathcal{V}_1$ is $\mathcal{V}$ itself this rule transforms into the usual implicit reflection rule $IRR(\mathcal{V})$ for $\mathcal{V}$.

A better internally verifiable theoretical description of the verification scheme above is given in terms of explicit reflection in Section 7.

Another class of examples that need reflection has been shown to me by Robert Constable and Stuart Allen. These examples are provided by completeness theorems and other provability tests which play important role in theoretical logic and mathematics and which are now within the scope of interests of advanced automated deduction systems.

Definition 9. *A* **provability test** *for $\mathcal{V}$ is a formula TEST(x) such that $\mathcal{V} \vdash$ TEST($\ulcorner\varphi\urcorner$) $\to \Box\varphi$ for any formula φ. $\mathcal{V}$ is* **stable with respect to provability tests** *if for every provability test TEST(x) and every formula φ*

$$\mathcal{V} \vdash \mathit{TEST}(\ulcorner\varphi\urcorner) \text{ yields } \mathcal{V} \vdash \varphi.$$

. In other words, if φ passes a provability test then φ is provable in $\mathcal{V}$.

Example 4. Formalized completeness and decidability theorems may be regarded as provability tests. Indeed, a completeness theorem generally states that if φ is valid with respect to a certain semantics then φ is provable. In its formalized version such a theorem is a formula *VALID*($\ulcorner\varphi\urcorner$)$\to \Box\varphi$ provable in $\mathcal{V}$, and one can take *VALID(x)* as *TEST(x)*. A formalized decidability theorem usually has a form $\mathcal{V} \vdash$ *TEST*($\ulcorner\varphi\urcorner$) $\leftrightarrow \Box\varphi$, where *TEST(x)* is a formula describing the decidability algorithm working on the code of φ and deciding whether φ is provable.

Theorem 2. *$\mathcal{V}$ is stable with respect to provability tests iff the implicit reflection rule IRR($\mathcal{V}$) is admissible in $\mathcal{V}$.*

Proof. Let *IRR*($\mathcal{V}$) be admissible in $\mathcal{V}$, and *TEST(x)* be a provability test. If $\mathcal{V} \vdash$ *TEST('φ')* then $\mathcal{V} \vdash \Box\varphi$ and, by the reflection rule, $\mathcal{V} \vdash \varphi$. Therefore $\mathcal{V}$ is stable with respect to provability tests.

Let now $\mathcal{V}$ be stable with respect to all provability tests. In particular, the stability with respect to the trivial test when *TEST(x)* is *Provable(x)* means that $\mathcal{V} \vdash \Box\varphi$ yields $\mathcal{V} \vdash \varphi$ for every φ, i.e. *IRR*($\mathcal{V}$) is admissible in $\mathcal{V}$.

5 Metamathematical Cost of Soundness and Stability

In this section we will find lower and upper bounds for the minimal metatheory $\mathcal{M}$ capable of establishing soundness and stability of a given verification system $\mathcal{V}$.

We will use the Turing progression as the standard scale to measure the metamathematical strength of a given extension of the basic theory ([13]). The Turing progression $\mathcal{V}^c_\alpha$ of theories (cf. [18], [10], [2]) for $\mathcal{V}$ is obtained from $\mathcal{V}$ by iterating the consistency assumptions along the Church-Kleene system of constructive ordinals α.

We consider the first ω theories from the Turing progression.

$$\mathcal{V}^c_0 = \mathcal{V}, \qquad \mathcal{V}^c_{n+1} = \mathcal{V}^c_n + \mathit{Consis}(\mathcal{V}^c_n), \qquad \mathcal{V}^c_\omega = \bigcup_n \mathcal{V}^c_n.$$

If $\mathcal{V}$ is correct with respect to the standard model of arithmetic, then the following strict inclusions hold:

$$\mathcal{V}_0^c \subset \mathcal{V}_1^c \subset \mathcal{V}_2^c \subset \ldots \subset \mathcal{V}_\omega^c.$$

Soundness was described in [8] as the condition that "We must be entirely convinced that any proof of a theorem which the system certifies as correct should indeed be so." A straightforward way to formalize soundness would be to assume some sort of the semantics for $\mathcal{V}$, to take $\mathcal{M}$ powerful enough to express the notion of truth for the $\mathcal{V}$-formulas and to establish inside $\mathcal{M}$ a formal analogue of the statement

for every sentence φ if φ is provable then φ is true.

This approach would require a fairly strong $\mathcal{M}$. In particular, one needs to extend the language of $\mathcal{V}$ in order to write down formulas *"φ is true"*; by the well-known Tarski theorem there is no such formula in the language of $\mathcal{V}$ itself.

In fact, in the proof checking context, a verification system $\mathcal{V}$ deals with the true values of formal statements of an especially simple type, namely provable Δ_1 sentences $[\![t]\!]\varphi$. In this paper we assume that soundness of a verification system $\mathcal{V}$ means that all sentences $[\![t]\!]\varphi$ derivable in $\mathcal{V}$ are true.

Theorem 3. *1. $\mathcal{V}$ is consistent iff $\mathcal{V} \vdash [\![t]\!]\varphi$ implies $[\![t]\!]\varphi$,*
2. $\mathcal{V}$ suffices to establish 1.

Proof. If for all φ $\mathcal{V} \vdash [\![t]\!]\varphi$ implies $[\![t]\!]\varphi$, then no false sentences of the kind $[\![t]\!]\varphi$ is provable in $\mathcal{V}$, therefore $\mathcal{V}$ is consistent.

Suppose $\mathcal{V}$ is consistent and let $\mathcal{V} \vdash [\![t]\!]\varphi$. If $[\![t]\!]\varphi$ were false, then $\mathcal{V} \vdash \neg[\![t]\!]\varphi$, by Δ_1 completeness of $\mathcal{V}$. This leads to a contradiction in $\mathcal{V}$.

2. The straightforward formalization of the proof of 1 with the use of provable Δ_1 completeness of $\mathcal{V}$. ∎

Corollary 1. *Simple consistency of $\mathcal{V}$ is necessary and sufficient for soundness of a verification system $\mathcal{V}$.*

Now we will figure out what metatheory can establish stability.

Definition 10. *By* $\mathcal{V}$ **is stable** *we understand the $\mathcal{V}$-formula which is the natural formalization of the stability property of $\mathcal{V}$. By* **implicit reflection rule is admissible in** $\mathcal{V}$ *we mean the natural formalization in the language of $\mathcal{V}$ of the property that IRR($\mathcal{V}$) is admissible in $\mathcal{V}$; we will denote this formula*

$$\forall x(\Box\Box x \to \Box x).$$

Theorem 4.

$$\mathcal{V} \ \vdash \ \text{“}\mathcal{V}\ \textbf{is stable} \leftrightarrow \textbf{implicit reflection rule is admissible in}\ \mathcal{V}\text{”}$$

Proof. The straightforward (though delicate) formalization of the proof of Theorem 1. ∎

Theorem 5. *Stability of an ω-consistent verification system is not provable in this system.*

Proof. By Theorem 4, stability is provable in $\mathcal{V}$ iff $\mathcal{V} \vdash \forall x(\Box\Box x \to \Box x)$. Let x is the code of $\perp$. Then $\mathcal{V} \vdash \Box\Box\perp \to \Box\perp$. By Löb's theorem, $\mathcal{V} \vdash \Box\perp$, which is impossible for an ω-consistent $\mathcal{V}$. ∎

It follows from the above that the minimal metatheory for soundness and implicit stability is

$$\mathcal{M} = \mathcal{V} + \mathit{Consis}(\mathcal{V}) + \forall x(\Box\Box x \to \Box x).$$

Theorem 6. *If $\mathcal{V}$ is correct with respect to the standard model of arithmetic then the metatheory for soundness and implicit stability strictly subsumes the first ω steps of the Turing progression.*

Proof. In order to establish $\mathcal{V}^c_\omega \subset \mathcal{M}$ consider the formulas $\Box^0\perp = \perp$, $\Box^{n+1}\perp = \Box(\Box^n\perp)$. First of all we note that under the assumptions made about $\mathcal{V}$ the formula $\mathit{Consis}(\mathcal{V}^c_n)$ is provably equivalent in $\mathcal{V}$ to $\neg\Box^{n+1}\perp$ (cf. [2]). Indeed, $\mathit{Consis}(\mathcal{V}^c_0)$ is $\mathit{Consis}(\mathcal{V})$, i.e. $\neg\Box\perp$. Then $\mathit{Consis}(\mathcal{V}^c_1)$ is a formula stating that $\mathcal{V} + \mathit{Consis}(\mathcal{V}) \nvdash \perp$, i.e. $\mathcal{V} + \neg\Box\perp \nvdash \perp$. This is equivalent to $\mathcal{V} \nvdash \neg\Box\perp \to \perp$ and $\mathcal{V} \nvdash \Box\perp$. Therefore, $\mathit{Consis}(\mathcal{V}^c_1)$ is equivalent to $\neg\Box\Box\perp$. Similar argument works for $n = 2, 3, 4, \ldots$.

Now we show how to derive all $\neg\Box^n\perp$, $n = 1, 2, 3, \ldots$ in $\mathcal{M}$. The case $n = 1$ is covered by the assumption that $\mathcal{M} \vdash \mathit{Consis}(\mathcal{V})$, which is equivalent to $\mathcal{M} \vdash \neg\Box\perp$, or $\mathcal{M} \vdash \Box\perp \to \perp$ For $n = 2$ put $x = \perp$ in $\forall x(\Box\Box x \to \Box x)$. Then $\mathcal{M} \vdash \Box\Box\perp \to \Box\perp$. Since we have already had $\mathcal{M} \vdash \Box\perp \to \perp$, we conclude that $\mathcal{M} \vdash \Box\Box\perp \to \perp$, i.e. $\mathcal{M} \vdash \neg\Box\Box\perp$. A similar argument works for $n = 3, 4, 5, \ldots$. Thus

$$\mathcal{V}^c_\omega \subset \mathcal{M}.$$

Now we will check that $\mathcal{V}^c_\omega \neq \mathcal{M}$. Suppose

$$\mathcal{V}^c_\omega \vdash \mathit{Consis}(\mathcal{V}) \wedge \forall x(\Box\Box x \to \Box x).$$

By the compactness argument, there is a natural number n such that

$$\mathcal{V}^c_n \vdash \mathit{Consis}(\mathcal{V}) \wedge \forall x(\Box\Box x \to \Box x).$$

Since $\mathcal{V}^c_\omega \subset \mathcal{M}$, $\mathcal{M}$ proves the consistency of $\mathcal{V}^c_n$. Therefore

$$\mathcal{V}^c_n \vdash \mathit{Consis}(\mathcal{V}^c_n),$$

which is impossible by the second Gödel incompleteness theorem for $\mathcal{V}^c_n$. ∎

6 Metamathematical Cost of Implicit Reflection

In an ω-consistent verification system $\mathcal{V}$ the rule of implicit reflection $IRR(\mathcal{V})$ is admissible, i.e. $\mathcal{V} \vdash \Box\varphi$ yields $\mathcal{V} \vdash \varphi$ for any formula φ. The most simple formalization of the admissibility property is the scheme $\Box\Box\varphi \rightarrow \Box\varphi$, where $\Box\psi$ stands for the formula of provability of ψ in $\mathcal{V}$. A general procedure of incorporating implicit reflection rule into a verification system $\mathcal{V}$ may be presented by the following *reflection tower* of extensions of $\mathcal{V}$ (cf. [12], [1], [11]):

$$\mathcal{V}^r_0 = \mathcal{V},\ \ \mathcal{V}^r_{\alpha+1} = \mathcal{V}^r_\alpha + IRR(\mathcal{V}^r_\alpha),\ \ \mathcal{V}^r_\gamma = \bigcup_{\beta\prec\gamma} \mathcal{V}^r_\beta \text{ for a limit ordinal } \gamma.$$

For the sake of simplicity we assume in this section that $\mathcal{V}$ is sound with respect to the standard model of arithmetic.

In this section we will try to figure out what natural metatheory is able to establish the admissibility of all the reflection rules from the reflection tower.

Definition 11. *Implicit reflection principle $IRP(\mathcal{V})$ for a given system $\mathcal{V}$ is the scheme of formulas*

$$\{\Box\varphi\rightarrow\varphi \mid \ \varphi \text{ is a sentence of } \mathcal{V}\}.$$

Let us consider *Feferman's progression* of extensions of $\mathcal{V}$ by the implicit reflection principles ([10]):

$$\mathcal{V}^p_0 = \mathcal{V},\ \ \mathcal{V}^p_{\alpha+1} = \mathcal{V}^p_\alpha + IRP(\mathcal{V}^p_\alpha),\ \ \mathcal{V}^p_\gamma = \bigcup_{\beta\prec\gamma} \mathcal{V}^p_\beta \text{ for a limit ordinal } \gamma.$$

The system $\mathcal{V}^p_1$ proves admissibility of implicit reflection in $\mathcal{V}^r_0$, i.e. the scheme of formulas $\Box\Box\varphi\rightarrow\Box\varphi$. In addition $\mathcal{V}^p_1 \subset \mathcal{V}^r_1$, since every instance of the rule $\Box\varphi/\varphi$ in a proof in $\mathcal{V}^r_1$ can be emulated by the axiom $\Box\varphi\rightarrow\varphi$. Moreover, the inclusion $\mathcal{V}^p_1 \subset \mathcal{V}^r_1$ can be established in $\mathcal{V}$. Iterating this argument one can show that $\mathcal{V}^p_{\alpha+1}$ is the theory capable of establishing admissibility of the implicit reflection rule for $\mathcal{V}^r_\alpha$.

How bad really is the reflection tower for $\mathcal{V}$? The natural metatheory capable of verifying the whole reflection tower is the limit of Feferman's progression $\mathcal{V}^p_\alpha$ for all constructive ordinals α.

Proposition 1. *([10]) The limit of $\mathcal{V}^p_\alpha$ for all constructive ordinals α equals*

$$\mathcal{V}\ +\ \textit{all true } \Pi_1\textit{-sentences.}$$

It follows from the above that the natural metatheory for the reflection tower is not computably enumerable, and could not possibly be verified by any sound mathematical means. It contains, for example, the consistency statements for all consistent axiomatic theories, among them $Consis(\mathcal{ZF})$ (provided $\mathcal{ZF}$ is consistent).

In the next section we describe explicit reflection, which is internally verifiable and thus circumvents the reflection tower.

7 Explicit Reflection for Verification Systems

An alternative way to represent provability in a logical setting has been suggested in [3] – [6], where a basic theory of *explicit provability* was developed. The key idea of this approach is to switch from the uniform but implicit presentation of provability of φ as $\Box\varphi$ to a presentation of provability of φ by by a certain family of explicit *proof terms* $[\![t]\!]\varphi$ (i.e. $Proof(t, \ulcorner\varphi\urcorner)$) depending on the context. As it was shown in [5] and [6], every propositional property of the provability operator (to the extend of the modal logic **S4**) can be represented by the family of finitely generated proof terms. Within this explicit provability approach some old problems in theoretical logic were solved. In particular, explicit provability provided the intended provability semantics for intuitionistic logic by formalizing Brouwer-Heyting-Kolmogorov semantics (the problem was open since 1930) and for the modal logic **S4** (Goedel's problem, was open since 1933).

In this paper we introduce a reflection mechanism based on explicit provability. This mechanism could help to avoid metamathematical costs of using the implicit reflection without restricting real verification capacities of a system.

Theorem 7. *For every sentence φ such that $\mathcal{V} \vdash \varphi$ there is a ground term t of $\mathcal{V}$ such that $\mathcal{V} \vdash [\![t]\!]\varphi$.*

Proof. Given $\mathcal{V} \vdash \varphi$ let D be a derivation of φ in $\mathcal{V}$. Let *t='D'=rep(D)*. By the assumptions on the function *rep*, $[\![t]\!]\varphi$ holds. Since $\mathcal{V}$ is able to represent all true Δ_1 facts, $\mathcal{V} \vdash [\![t]\!]\varphi$.

Definition 12. *The* explicit reflection principle $ERP(\mathcal{V})$ *is the scheme of formulas $[\![t]\!]\varphi \to \varphi$ for all sentences φ and all ground terms t.*

Theorem 8. (Provability of explicit reflection [3]). *For any ground term t and formula φ*

$$\mathcal{V} \vdash [\![t]\!]\varphi \to \varphi.$$

Proof. We give a constructive proof of this lemma which delivers an algorithm for constructing a derivation of $[\![t]\!]\varphi \to \varphi$ in $\mathcal{V}$ given φ and t. First of all, by the proof checking procedure we calculate the truth value of $[\![t]\!]\varphi$. If this value is *TRUE*, then the ground term t represents a derivation of φ, from which by a straightforward reconstruction, we obtain the proof of $[\![t]\!]\varphi \to \varphi$. If the proof checker on $[\![t]\!]\varphi$ returns *FALSE*, then by the corresponding procedure we get the proof of $\neg[\![t]\!]\varphi$ in $\mathcal{V}$. From that by the straightforward transformation, we get the proof of $[\![t]\!]\varphi \to \varphi$. ∎

Corollary 2. *There is an algorithm which given a formula φ and a ground term t returns the ground term p such that*

$$\mathcal{V} \vdash [\![p]\!]([\![t]\!]\varphi \to \varphi).$$

Definition 13. *A rule Γ/φ is explicitly verifiable in $\mathcal{V}$ if there is a total computable function f such that $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$.*

Theorem 9.

1. *Every derivable and every provable rule is explicitly verifiable.*
2. *Every explicitly verifiable rule is verifiable.*
3. *Every explicitly verifiable rule is admissible.*

Proof. 1. There is a straightforward function behind every internal rule Δ/ψ which calculates the code of a proof of ψ given the codes of proofs of Δ. A natural formalization of this function in $\mathcal{V}$ gives a term f such that $\mathcal{V} \vdash [\![y]\!]\Delta \rightarrow [\![f(y)]\!]\psi$. The same holds for all derivable rules. If a rule Γ/φ is provable then $\mathcal{V} \vdash \Gamma \rightarrow \varphi$. By explicit formalization (Theorem 7), $\mathcal{V} \vdash [\![t]\!](\Gamma \rightarrow \varphi)$ for some ground term t. Let "$\cdot$" be a total and computable "application" function on proof codes, specified by the condition

$$\mathcal{V} \vdash [\![x]\!](\varphi \rightarrow \psi) \rightarrow ([\![y]\!]\varphi \rightarrow [\![x \cdot y]\!]\psi)$$

(cf. [5], [6]). In particular,

$$\mathcal{V} \vdash [\![t]\!](\Gamma \rightarrow \varphi) \rightarrow ([\![x]\!]\Gamma \rightarrow [\![t \cdot x]\!]\varphi),$$

which yields $\mathcal{V} \vdash [\![x]\!]\Gamma \rightarrow [\![t \cdot x]\!]\varphi$.

2. From $\mathcal{V} \vdash [\![y]\!]\Gamma \rightarrow [\![f(y)]\!]\varphi$ it easily follows that $\mathcal{V} \vdash \Box\Gamma \rightarrow \Box\varphi$.

3. Let $\mathcal{V} \vdash [\![y]\!]\Gamma \rightarrow [\![f(y)]\!]\varphi$ and suppose that $\mathcal{V} \vdash \Gamma$. By Theorem 7, $\mathcal{V} \vdash [\![t]\!]\Gamma$ for some ground term t. Therefore $\mathcal{V} \vdash [\![t]\!]\Gamma \rightarrow [\![f(t)]\!]\varphi$ and $\mathcal{V} \vdash [\![f(t)]\!]\varphi$. By Theorem 8, $\mathcal{V} \vdash [\![f(t)]\!]\varphi \rightarrow \varphi$. Thus $\mathcal{V} \vdash \varphi$ ∎

Definition 14. *The explicit reflection rule $ERR(\mathcal{V})$ is the rule $[\![t]\!]\varphi/\varphi$ for all ground terms t and all sentences φ.*

Theorem 10. *The explicit reflection rule $ERR(\mathcal{V})$ is explicitly verifiable in $\mathcal{V}$.*

Proof. By Theorem 8, $\mathcal{V} \vdash [\![p]\!]([\![t]\!]\varphi \rightarrow \varphi)$ for some ground term p. By the same argument about computable "application" as in the proof of the previous theorem,

$$\mathcal{V} \vdash ([\![y]\!][\![t]\!]\varphi \rightarrow [\![p \cdot y]\!]\varphi).$$

∎

Corollary 3. *The explicit reflection rule $ERR(\mathcal{V})$ is admissible for every verification system $\mathcal{V}$.*

Definition 15. *An extension $\mathcal{V}'$ of $\mathcal{V}$ is verifiably equivalent to $\mathcal{V}$ if there is a computable function g of $\mathcal{V}$ such that $\mathcal{V} \vdash [\![x]\!]'\psi \rightarrow [\![g(x)]\!]\psi$, where $[\![x]\!]'\psi$ stands for the formula "x is a proof of ψ in $\mathcal{V}'$. In other words, for a verifiably equivalent extension $\mathcal{V}'$ there is an algorithm that transforms proofs in $\mathcal{V}'$ into proofs of the same facts in $\mathcal{V}$.*

Theorem 11. *An extension of a verification system by an explicitly verified rule is verifiably equivalent to the original system.*

Proof. Let a rule Γ/φ be explicitly verifiable in a verification system $\mathcal{V}$, i.e. there is a computable function f such that $\mathcal{V} \vdash [\![y]\!]\Gamma \rightarrow [\![f(y)]\!]\varphi$. Let $\mathcal{V}'$ be $\mathcal{V} + \Gamma/\varphi$. The function $g(x)$ works as follows. It travels along the proof tree in $\mathcal{V}'$ coded by x and calculates the code of a proof tree in $\mathcal{V}$ of the same sentence (sequent). If the observed node is a leaf node, then it corresponds to an axiom of $\mathcal{V}'$, which is an axiom of $\mathcal{V}$ as well. In this situation g does not change the the proof at all.

Let the observed node correspond to an application of an internal rule Δ/θ, and let $\boldsymbol{u}$ be the values of g on the predecessors of the current node, i.e. $\mathcal{V} \vdash [\![\boldsymbol{u}]\!]\Delta$. By Theorem 9, there is a computable function h such that $\mathcal{V} \vdash [\![\boldsymbol{y}]\!]\Delta \rightarrow [\![h(\boldsymbol{y})]\!]\theta$. Substituting u for y we derive $[\![h(\boldsymbol{u})]\!]\theta$ in $\mathcal{V}$. Let g map the observed node to $h(\boldsymbol{u})$.

Let the observed node correspond to an application of the new rule Γ/φ, and let $\boldsymbol{v}$ be the values of g on the predecessors of this node, i.e. $\mathcal{V} \vdash [\![\boldsymbol{v}]\!]\Gamma$. By the conditions of the theorem $\mathcal{V} \vdash [\![\boldsymbol{y}]\!]\Gamma \rightarrow [\![f(\boldsymbol{y})]\!]\varphi$. Substitute v's for y's, conclude that $\mathcal{V} \vdash [\![f(\boldsymbol{v})]\!]\varphi$ and let g map the observed node to $f(\boldsymbol{v})$.

Eventually, at the root node of the $\mathcal{V}'$-proof (coded by) x the function g returns the code of a $\mathcal{V}$-proof of the formula (sequent) previously proven by x. ∎

8 Practical Suggestions

As one can see, explicit reflection avoids some of the troubles inherent in implicit reflection. Here is the list of practical suggestions for the designers of verification systems.

1. **Proof Checking.** Explicit reflection is used by default in proof checking when one concludes that $\mathcal{V}$ has verified a fact φ given that $\mathcal{V} \vdash [\![t]\!]\varphi$ for some proof code t. This scheme is theoretically correct and does not contain any extra hidden metamathematical costs. Here the use of explicit reflection should be acknowledged.

2. **Extendable Verification Systems.** Here the use of explicit reflection may be twofold. Firstly, it appears in the *assertion insertion mode* (cf. [8]), when it is established that $\mathcal{V} \vdash [\![t]\!]\varphi$ and then φ is stored as a verified fact (i.e. a new axiom) of $\mathcal{V}$. We have nothing specific to add here, since this mode as presented above (and in [8]) already agrees with the explicit reflection recommendations. Secondly, the explicit reflection appears in the *rule insertion mode*, when Γ/φ is verified in $\mathcal{V}$ and then added to $\mathcal{V}$ as a new inference rule. The explicit reflection suggests verifying the rule Γ/φ in $\mathcal{V}$ explicitly, i.e. by constructing a computable function f such that $\mathcal{V} \vdash [\![y]\!]\Gamma \rightarrow [\![f(y)]\!]\varphi$. By doing this we guarantee that the resulting extension is verified in the old system without any hidden meta assumptions.

> If the rule insertion mode uses explicit verification only, then there is no need to have a special built-in reflection mechanism: provable stability of the system is preserved by explicit verification (Theorem 11).

Interestingly enough, there are substantial classes of verification systems where the implicit verification in a certain sense yields the explicit one. For example, in many intuitionistic systems $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$ implies $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$ for some computable function f (cf. [17]). However, the proof of this fact itself cannot be formalized in $\mathcal{V}$ and its use in the rule insertion mode leads to some sort of a reflection tower. Therefore, ever for the constructive systems the practical suggestion would be to use the explicit verification, i.e. to establish $\mathcal{V} \vdash [\![y]\!]\Gamma \to [\![f(y)]\!]\varphi$ directly rather than to prove $\mathcal{V} \vdash \Box\Gamma \to \Box\varphi$ and then to apply a general theorem of obtaining the explicit verification from the implicit one; this involves some hidden and potentially high metamathematical costs.

3. **Advanced systems with built-in reflection mechanisms.** There is a number of systems which have or intend to have such mechanisms. The paper [11] mentions several of them: **FOL**, **NQTHM**, **HOL** and **Nuprl**. At least one more is coming: **MetaPrl** at Cornell University. Probably more systems will join this set since reflection arguments are often used in mathematical and common reasoning (cf. Section 4). The existing implicit reflection mechanisms in these systems lead to unnecessary metamathematical costs (cf. Section 6). For such systems the idea of having explicit reflection (perhaps, along with the implicit one) might be seriously considered, because the explicit reflection can be added to a system without any extra metamathematical assumptions at all (Theorem 10).

9 Acknowledgements

I am indebted to Robert Constable for attracting my attention to the problem of reflection in verification systems and for support during my work on this paper. I am also grateful to Stuart Allen, Anil Nerode, Elena Nogina and Vaughan Pratt for valuable suggestions and criticism. Many thanks to Martin Davis for fruitful discussion and for sending me a copy of his paper.

References

1. Allen, S., Constable R., Howe D., Aitken W.: The Semantics of Reflected Proofs. In: Proceedings of the Fifth Annual Symposium on Logic in Computer Science, Los Alamitos, CA, USA, IEEE Computer Society Press (1990) 95-107.
2. Artemov, S.: Extensions of Theories by the Reflection Principles and the Corresponding Modal Logics. Ph.D. Thesis (in Russian). Moscow (1979)
3. Artemov, S., Strassen, T.: The Basic Logic of Proofs. In.: Lecture Notes in Computer Science. Vol. 702. Springer-Verlag (1993) 14-28.
4. Artemov, S.: Logic of Proofs. Annals of Pure and Applied Logic **67** (1994) 29-59
5. Artemov, S.: Operational Modal Logic. Technical Report MSI 95-29, Cornell University (1995)
6. Artemov, S.: Explicit Provability: the Intended Semantics for Intuitionistic and Modal Logic. Technical Report CFIS 98-10, Cornell University (1998)
7. Boolos, G.: The Unprovability of Consistency: An Essay in Modal Logic. Cambridge University Press (1979)
8. Davis, M., Schwartz, J.: Metamathematical Extensibility for Theorem Verifiers and Proof Checkers. Computers and Mathematics with Applications **5** (1979) 217-230

9. Feferman, S.: Arithmetization of Metamathematics in a General Setting. Fundamenta Mathematicae **49** (1960) 35-92
10. Feferman, S.: Transfinite Recursive Progressions of Axiomatic Theories. Journal of Symbolic Logic **27** (1962) 259-316
11. Harrison, J.: Metatheory and Reflection in Theorem Proving: A Survey and Critique. University of Cambridge (1995) `http://www.dcs.glasgow.ac.uk/ tfm/hol-bib.html#H`
12. Knoblock, T., Constable, R.: Formalized Metareasoning in Type Theory. In: Proceedings of the First Annual Symposium on Logic in Computer Science. Cambridge, MA, USA, IEEE Computer Society Press (1986) 237-248
13. Kreisel, G., Levy, A.: Reflection Principles and Their Use for Establishing the Complexity of Axiomatic Systems. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **14** (1968) 97-142
14. Pollack, R.: On Extensibility of Proof Checkers. Lecture Notes in Computer Science, Vol. 996. Springer-Verlag, Berlin Heidelberg New York (1995) 140-161
15. Smorynski, C.: The Incompleteness Theorems. In: Barwise, J (ed.): Handbook of Mathematical Logic, Vol. 4. North-Holland, Amsterdam (1977) 821-865
16. C. Smorynski, C.: Self-Reference and Modal Logic. Springer-Verlag, Berlin (1985)
17. A.S. Troelstra, A.S., van Dalen, D.: Constructivism in Mathematics. An Introduction, Vol. 1, Amsterdam; North Holland (1988)
18. Turing, A.: Systems of Logics Based on Ordinals. Proceedings of the London Mathematical Society, Ser. 2, Vol. 45 (1939) 161-228